RESEARCH MEMORANDUM 76-10

TRANSL--THE PLANIT TRANSLATOR PROGRAM: INSTALLATION AND APPLICATION

Charles H. Frye
The Northwest Regional Educational Laboratory

Educational Technology and Training Simulation Technical Area

U. S. Army Research Institute for the Behavioral and Social Sciences

May 1976

DISPOSITION FORM

For use of this form, see AR 340-15, the proponent agency is TAGCEN.

REFERENCE OR OFFICE SYMBOL	SUBJECT
PERI-TP	Clearance and Transmittal of Reports to DTIC

TO	FROM	DATE
DDC-DAA-1 ATTN: Mr. Schrecengost	ARI Rsch Pub Group	8 Nov 79 Ms Price/48913

1. The reports listed on Inclosure 1 are approved for public release with unlimited distribution (50 numbered ARI Research Memorandums, 74-1 thru 76-30).

2. These are among the previously unrecorded ARI reports which you identified t us 22 June 1979 as not in your retrieval system. The accompanying box contains at least one copy of each report for your retention and reproduction.

Helen S. Price

HELEN S. PRICE
Research Publications Group
Army Research Institute

1 incl
List of reports, 1974-76

x 4

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or special
A

DA FORM 1 FEB 62 2496 REPLACES DD FORM 96, WHICH IS OBSOLETE.

(14) ARI-RM-76-10

(16) Army Project Number
20762717A721

Educational Technology and
Training Simulation
Contract Number DAHC 19-74-C-0038

(9) Research Memorandum 76-10

(6) TRANSL--THE PLANIT TRANSLATOR PROGRAM: INSTALLATION AND APPLICATION

(10) Charles H. Frye
The Northwest Regional Educational Laboratory

Submitted by:
James D. Baker, Chief
Educational Technology and Training Simulation Technical Area

(11) May 1976

(12) 78

Approved by:

Joseph Zeidner, Director
Organizations and Systems Research
Laboratory

J. E. Uhlaner, Technical Director
U.S. Army Research Institute for
the Behavioral and Social Sciences

---

*(Programming language for interactive teaching)

**AN/GYK-12

ABSTRACT

This document contains full information for installing and operating a program which is designed to translate the FORTRAN from the PLANIT* system of programs into the TACPOL language for compilation on the ~~ANGYK-12~~** computer.

PLANIT is an interactive training system which is now operational on the ~~ANGYK-12~~**, having been installed via the program herein described. While the normal method for installing the PLANIT system requires the existence of a FORTRAN compiler on the target computer, this was circumvented for the ~~ANGYK-12~~** installation by developing the Translator (TRANSL) program to first translate the FORTRAN into TACPOL for use with the PSS-B compiler, the only compiler on the ~~ANGYK-12~~**.

Although the installation of PLANIT is complete on the ANGYK-12** computer, future updates of PLANIT may necessitate additional use for the TRANSL program.

BACKGROUND OF THE PLANIT USER TRAINING SYSTEM

Several explicit user requirements converged to generate the research which resulted in the documents contained in this set of reports. The need for some type of user training subsystem in support of tactical automatic data processing (ADP) system developments was clearly established during the evolutionary phase of the Army Tactical Operations System (TOS) development in Europe.[1] In 1974, after a decade of involvement in the development of tactical ADP systems, the Army Computer Systems Command summarized this experience into six "Lessons Learned."[2] One of these lessons was: A dedicated and trained user is required if tactical ADPS is to succeed.

One approach toward meeting this requirement is to apply techniques derived from modern educational technology and the computer sciences by embedding training subsystem packages within the operating system and then using the system itself to teach the user how to use the system. The approach was delineated in a concept paper,[3] which was subsequently submitted, evaluated and found by key Army Personnel to have merit. As a consequence, a requirement was placed on the Army's Behavior and Systems Research Laboratory (BESRL--the predecessor of what is now the Army Research Institute) by what was then the Assistant Chief of Staff for Force Development (ACSFOR) and the Director of Army Research, Office of the Chief of Research and Development (OCRD),[4,5] to effectuate the research necessary to test the concept.

---

[1]Baker, J. D. "Human Factors Experimentation Within a Tactical Operations System (TOS) Environment." Proceedings: Office of Naval Research Sponsored Tri-Service Coordination Meeting, London, England, 20-21 February 1968.

[2]Memorandum from Headquarters, U.S. Army Computer Systems Command to Assistant Deputy Commander, CACDA, Ft. Leavenworth, KA; Deputy Commander, MASSTER, Fort Hood, TX; Project Manager, Army Tactical Data Systems, Fort Monmouth, NJ, dtd 30 January 1974, Subject: TSDG Lessons Learned.

[3]Memorandum from U.S. Army Behavior and Systems Research Laboratory to Assistant Chief of Staff for Force Development, dated 28 September 1971, Subject: Proficiency Maintenance Using Computer-Assisted Instruction in an Operational Setting.

[4]Memorandum from Assistant Chief of Staff for Force Development to Chief of Research and Development, dated 10 November 1971; with 18 November 1971 indorsement to Behavior and Systems Research Laboratory, Subject: Request for Research in Application of Tactical Data Systems for Training.

[5]Memorandum from Chief of Research and Development to Assistant Chief of Staff for Force Development, dated 29 Nov 1971, Subject: Request for Research in Application of Tactical Data Systems for Training.

The terms of the requirement actually levied, however, went well beyond the scope of the original concept and called for a simultaneous attack on all facets of the problem associated with testing the feasibility of the approach. In terms of broadened scope, the primary role of these systems is in support of tactical operations. Our original concept paper suggested a potential, select secondary role for these computerized tactical data systems, viz., that of directly supporting the system user by using the system itself, in a stand-alone mode, to teach the user how to use the system. The agencies structuring the research requirements saw a possible tertiary role for these systems. About the time they were structuring their requirements, the Army's Dynamic Training Board identified the maintenance of proficiency of Military Occupation Specialty (MOS) 11B40, the light weapons infantryman, as a glaring unit training problem and suggested that Computer-Assisted Instruction (CAI) as one technique for alleviating the situation.[6] In addition, a subsequent Continental Army Command (CONARC) Task Group report on CAI identified the 11B40 MOS as a top contender for attention in the "non-technical" skills area.[7] Consequently, the scope of the effort was expanded to encompass an examination of a tertiary role, i.e., in support of the system's parent unit by using these computers to meet individual and unit training requirements such as those associated with the 11B40 MOS. Additionally, in response to concern that the implementation of the Modern Volunteer Army concept might produce a need for general education development (GED) upgrading it was determined that an examination should be made of the feasibility of employing extant CAI GED on tactical computers in an operational setting. The assumption was made that accomplishment of these latter requirements would be tantamount to proving the feasibility of the secondary role concept as well. The test, therefore, would be a cost-effective undertaking since it would provide data directed toward answering a number of diverse questions concerned with a common training delivery system, viz., tactical computers.

Irrespective of whether it was the secondary or tertiary role concept being assessed, four major components were required: a test in a credible operational environment; appropriate hardware; functioning software and representative people-ware. The vehicle for this overall assessment was MASSTER[8] Test FM 122, "IBCS: Automated Instruction." The hardware was a "given" viz., the Developmental Tactical Operations

---

[6]Report of the Board for Dynamic Training, Volume II. 17 December 1971, page 116.

[7]Headquarters, United States Continental Army Command Task Group Report and Computer Assisted Instruction. April 1972.

[8]MASSTER - Modern Army Selected Systems Test, Evaluation, and Review--is the Army's test bed for assessing equipment, concepts and doctrine. This activity is located at Fort Hood, Texas.

System (DEVTOS) which was then located at Fort Hood, Texas (Hoyt, et al[9] provide a description of the hardware). Likewise, the people were a "given"--our student population would be MOS 11B40 personnel drawn from the 2nd Armored Division and 1st Cavalry Division located at Fort Hood. The question of what "software" approach to take (specifically, whether to use an existing student/author language) was key to the success or failure of Test 122. Clearly, the decision made at this juncture would determine whether we would hit the assigned "test window" in time to conduct the test. As a related issue, courseware development would largely depend upon the structure of the student/author language selected, so courseware development could not commence until this decision was made. The decision itself had to be correct and timely--and whatever decision was made would undoubtedly be risky.

To add to the difficulty in reaching a decision, it must be realized that it could not be made unilaterally. Conduct of a test of the complexity of MASSTER Test FM 122 required support from and coordination between a number of different agencies--key among them being mutual cooperation of the organization which had DEVTOS responsibility, the U.S. Army Computer Systems Command (USACSC), and the Army Research Institute (ARI). A Memorandum of Understanding[10] was drawn up between these two organizations and, as the first USACSC task in this joint undertaking, a MASSTER Test 122 CAI Concept Paper[11] was to provide alternative concepts for implementing automated instruction materials on the DEVTOS in support of MASSTER Test 122. Concurrent with this effort, a contract was let by ARI with the System Development Corporation (SDC) to develop the courseware (i.e., the instructional materials which would be presented through CAI). The first task SDC had to accomplish was to provide alternative student/author language alternatives for generating the courseware and to determine which alternative provided the best likelihood of success under the test conditions and time constraints imposed. In essence, the combined results of these analytic studies were expressed as follows: "At this stage, many alternative design concepts can be formulated. However, due to time constraints on the implementation of any concept, the only alternative concept considered feasible...is the use of PLANIT."[12]

---

[9]Hoyt, W. G., Butler, A. K. and Bennik, F. D. "Application of Tactical Data Systems for Training: DEVTOS Feasibility Determination and Selection of an Instructional Operating System." ARI Technical Paper 267, October 1975.

[10]Memorandum of Understanding Between Commander, U.S. Army Research Institute and Commander, U.S. Army Computer Systems Command, Dated 5 June 1973.

[11]Bunker-Ramo Technical Note "MASSTER Test 122--Computer Assisted Instruction (CAI) Concept Paper," February 1975, prepared for the U.S. Army Computer Systems Command.

[12]Ibid. 11, page 18.

PLANIT (Programming Language for Interactive Teaching) is an instructional system consisting of an author language and supporting computer programs for preparing, editing and presenting any subject matter suitable for individualized CAI presentation to students, as well as recording all relevant response data for immediate utilization and subsequent analyses. PLANIT was developed over an eleven year period under the aegis of the National Science Foundation (NSF) at a total investment cost of approximately $740,000. The main goal of this NSF project was to produce a student/author language which would be fully transportable and guaranteed compatible with a large and diversified class of machines.[13] We at ARI take professional pride in the fact that it was our early and subsequent work with PLANIT which validated this visionary transportability notion of NSF.[14] We also take "economic" pride in the fact that we capitalized upon an already "hefty" U.S. Government investment to solve a problem, rather than slipping into the classic mold of "reinventing the wheel" by starting from scratch and building a separate student/author language tailored to the hardware/software system constraints.

To lower the curtain on MASSTER Test FM 122, the test was successfully conducted and demonstrated that it was feasible to use tactical computers in a stand-alone training mode to satisfy individual and unit training requirements. It was found that automated instruction in a field setting was enthusiastically accepted by the non-commissioned officers (NCO's) examined and, as a training medium, it proved to be more effective than the traditional study-method of training.[15,16,17,18,19]

---

[13]Frye, C. H. "A Report on PLANIT: One Stage of Completion," Final Report for the National Science Foundation Grant No. EPP73-07319 A04, August 1975.

[14]For a complete account of the experiences of ARI in installing, using and evaluating PLANIT in an Army setting, including all the "warts and blemishes" uncovered during this endeavor, see: Johnson, C. "Implementation of PLANIT at the U.S. Army Research Institute for the Behavioral and Social Sciences," PLANIT Newsletter, July 1975.

[15]Hoyt, W. G. and Baker, J. D. The use of tactical computers to provide weapons and tactics training to combat NCO's: Results of a field test. Proceedings: Sixteenth Annual Conference of the Military Testing Association MTA, U.S. Coast Guard Institute, Oklahoma City, OK. 21-25 October 1974.

[16]Hoyt, W. G., Butler, A. K. and Bennik, F. D. Application of tactical data systems for training: Volume II - CAI/DEVTOS automation studies. ARI Technical Paper 267, October 1975.

[17]Hoyt, W. G., Butler, A. K. and Bennik, F. D. Application of tactical data systems for training: Volume I - Executive Summary. ARI Technical Paper ___, in preparation.

But the results of this test proved more than the preceding. They also indicated that the obvious Army needs mentioned at the outset of this preface, could be met by applying this technology to a real and present problem. It also went beyond the exploratory stage and satisfied a specific Army requirement. The U.S. Army Combat Developments Command (CDC)/Systems Analysis Group (now the U.S. Army Training and Doctrine Command/Combined Arms Combat Developments Activity, or TRADOC/CACDA) had levied the following requirement[20] on ARI:

> The Proposed Material Need for the Tactical Operations System - TOS (Unclassified title, portions of contents classified CONFIDENTIAL) states: "During system non-tactical employment the equipment shall have the capability to permit the training of user personnel without affecting the mission ready capability of the system." While the need exists, no specific data are extant which can be brought to bear on this problem. The requested research will provide data which could impact on all TOS users and result in considerable savings in training costs related to the user's need to maintain proficiency in the use of these systems.

The 122 Test data satisfied the CDC requirement. The Proposed Material Need (MN) for TOS was found to be a viable concept and that MN remains to this day as a bonafide component of the TOS program.

As previously discussed, the results from MASSTER Test FM 122 demonstrated the viability of the embedded training subsystem concept in general and that tactical data systems could be used in a tertiary role, i.e., specifically, that these systems could be used in a stand-alone mode in support of individual and unit softskills training requirements. But conceptually our main goal had always been to embed system specific training packages within the operating system itself and then to use the system to teach the user how to use the system--the earlier noted secondary role for these systems.

---

[18]Hoyt, W. G., Butler, A. K. and Bennik, F. D. Application of tactical data systems for training: Volume III - Development of courseware and analysis of results for MOS 11B40. ARI Technical Paper ___, in preparation.

[19]Hoyt, E. G., Butler, A. K. and Bennik, F. D. Application of tactical data systems for training: Volume IV - Development of courseware and analysis of results for GED math. ARI Technical Paper ___, in preparation.

[20]Letter, DARB-ARB 19 July 1972, Subject: New Research Requirements for the Human Resources Research and Development Program (RCS CSCRD 70 CRI); letter response from CDCSAG-AG1, same subject as above, dated 1 September 1972.

As a follow-on to Test 122, research was initiated under the aegis of the Product Manager, Computer Training Systems (PM CTS) through HRN 75-158 (and, subsequently, HRN 76-195) which tasked ARI to address the problem of reducing the novice user's difficulties by making tactical data systems (e.g., TOS[2], TACFIRE, TSQ-73, etc.) more "approachable" through applications of the embedded training concept.[21]

Because of its stage of development, the fact that its basic central processing unit would serve as the core for other Army Tactical Data Systems (ARTADS) to follow, and the fact that its operator training problems appeared to be amenable to reduction through the application of automated instructional technology, TACFIRE (the Army's field artillery tactical fire control system) was chosen by the PM CTS as the test vehicle for assessing the embedded training subsystems concept. The initial and specific requirements for the TACFIRE research were delineated in HRN 76-193, "Development and Evaluation of PLANIT Based Computer Embedded Training Packages for TACFIRE" which was prepared by personnel of the U.S. Army Field Artillery School, Fort Sill, OK.

Once again we were faced with the dilemma as to whether the best decision would be to develop a tailor-made student/author language smoothly fitted to the hardware/software constraints of the TACFIRE system, or to build upon our already successfully operating PLANIT system and attempt to install it on TACFIRE. The latter approach had many merits, among them: (1) it was an author language system with which we were familiar, while a customized system would be untested, costly and would require an extensive checkout; (2) a customized authoring system would be limited to a given TACFIRE configuration, whereas PLANIT would be transportable to the family of ARTADS systems, and (3) because of PLANIT's machine independent characteristics, courseware could be prepared on commercial computers and, after content checkout, easily installed on the tactical system, whereas a customized approach would tie-up the actual tactical system during courseware preparation.

The effort to install PLANIT on the AN/GYK-12 computer, the results of which are contained in this set of reports, was independently undertaken as Technology Based - Exploratory Development research and not as Advanced Development activity (i.e., it was not done in direct response to an explicit, stated user need). It serves as a classic example of what Dr. Malcolm R. Currie, Director of Defense Research and Engineering (DDR&E) was describing in the following statement to the Second Session of the 94th Congress: "The objective of the Technology Base is the advancement of technology applicable to future systems and subsystem

---

[21]Human Resource Need (HRN) 75-158, title: "User Training and Proficiency Maintenance in a Tactical Data Systems Environment," submitted as a research requirement for inclusion in the ARI FY 75 Advanced Development Work Program by the Product Manager, Computerized Training System, Fort Monmouth, NJ. HRN 76-195 was a revalidation of the requirements delineated in 75-158 for inclusion in the FY 76 Work Program.

options. These options (or new ideas) usually involve enhanced military capability, reduced cost, increased performance, better reliability and maintainability, more efficient use of resources or some combination of these attributes." Success in this effort would produce a broadly applicable, cost-effective vehicle for employing embedded training subsystem packages in a variety of military system settings.

It merits comment, however, that while this work was a Technology Based-Exploratory Effort, it had the potential for feeding into the Advanced Development program efforts associated with the user tasks presented in HRN 75-158, "User Training and Proficiency Maintenance in a Tactical Data Systems Environment," if the outcome were successful. Consequently, the PM-CTS was appraised of this effort at the outset and he, in turn, coordinated it with the Program Manager, Army Tactical Data Systems (PM ARTADS). During this coordination some valid points of criticism were raised[22] concerning the PLANIT approach. The PM ARTADS recommended that ARI meet with system developers, users and training agencies as soon as sufficient data were available to determine whether, or not, PLANIT would operate on TACFIRE. At that time a determination would be made concerning implementation implications and to assess if, indeed, this were the most effective approach to take, given the potential for impact on TACFIRE system development efforts. In keeping with this recommendation, a Workshop was convened at ARI in Arlington, VA on 1 October 1974 and these items were covered in detail with personnel from all of the suggested groups in attendance. The interaction was found to be most beneficial to all concerned and the concensus of the group was to install the system described in this set of reports on the TACFIRE system at Fort Sill, OK, and to use it as the test vehicle for assessing the embedded training concept on that ARTADS system.

This historic overview of the events leading up to the production of the set of quite specialized reports may seem untoward in view of the projected, limited set of users of these documents. It is, however, a quite meaningful forum for discussing these events. Too frequently the question is raised as to how did a particular research product originate and was it utilized. The intent here is to show that the warp and woof of concepts and coordination, requirements and research are so intertwined that a simple one-to-one relationship (one response, one use) does not tell the story--only a view of the whole cloth will put it into proper perspective. Additionally, it exemplifies a point made in the previously cited presentation by the Director of Defense Research and Engineering to the 94th Congress when he said: "To deploy systems DOD must not only pursue advanced technology but must endure the long years of research required to bring an idea through growth problems to a finished, proven and useful end product."

---

[22]Memorandum from Product Manager, Computer Training Systems (PM-CTS) to Program Manager, Army Tactical Data Systems (PM-ARTADS) 28 Jan 74, Subject: HRN 75-158 and 1st indorsement from PM-ARTADS to PM-CTS, same subject as above dated 7 February 74.

This set of reports provides detailed instructions for implementation and operation of PLANIT and auxiliary programs on the AN/GYK-12 computer. The set consists of a report on:

TRANSL - The PLANIT Translator Program: Installation and Application

PLANIT Support Programs - Operator/user manual

PLANIT Utility Program - Operator/user manual

PLANIT Support and Utility Programs - Test Procedure

PLANIT Support and Utility Programs - Flow Charts.

The first report contains the information for installing and operating a program which is designed to translate the FORTRAN from the PLANIT system of programs into the TACPOL language for compilation on the AN/GYK-12 computer. The second covers the general and specific aspects of leading and operating PLANIT on the AN/GYK-12 computer. The third document covers the general and specific aspects of operating the PLANIT utility programs which are a specialized group of routines developed to accomplish various tasks in support of the AN/GYK-12 computer installation of PLANIT. The fourth report covers the procedures used to verify that PLANIT Support and Utility Programs are functioning as per specifications. The fifth document provides the detailed flow charts of the computer logic of the PLANIT Support and Utility Programs.

The effort detailed in the first report (i.e., TRANSL) was accomplished under ARI Contract DAHC19-74-C-0038 by the Northwest Regional Educational Laboratory, Portland, Oregon. The other four reports in the series were prepared by the Data Systems Division, Litton Systems Inc., Van Nuys, CA under ARI Contract No. DAHC19-74-C-0064.

CONTENTS

FIGURES

APPENDICES

TRANSL -- THE PLANIT TRANSLATOR PROGRAM

INSTALLATION AND APPLICATION

INTRODUCTION

Soon after computers began to be used through time-shared, interactive typewriter terminals, experimental efforts were initiated to use this new facility for training purposes. Within a few years several systems were developed which made the computer more readily adaptable for this use. PLANIT (Programming Language for Interactive Teaching) is one such system. The user features and intended applications of PLANIT are documented elsewhere and will not be elaborated here. What is important is the fact that, unlike other such systems, PLANIT is a complete and fully portable time-sharing system suited particularly to training applications. These features have contributed to the selection of PLANIT for installation on a variety of military computers, some of which have non-standard features and lack the usual software support on which some interactive systems depend.

One such specialized military computer is the ANGYK-12 (or the Litton designation, L3050) which is designed to be used in several tactical systems applications such as the TACFIRE fire control system. Although the ANGYK-12 is a versatile and sophisticated computer, it lacks many of the facilities which one would expect to find on a commercial computer of comparable size. Instead of the usual menu of programming languages, it has only TACPOL (a PL-1 like language). The TACPOL language was designed especially for this computer. Although the similarities to conventional programming languages are many, there are some important differences such as the lack of any floating point numerical operations.

Because of the above considerations, PLANIT was found to be the only available training system which could be installed on the ANGYK-12 computer without extensive redevelopment efforts. However, PLANIT normally makes use of the FORTRAN IV language in the installation process to achieve its portability. Since FORTRAN IV was not available on the ANGYK-12, a special translator program was written which intercepts the FORTRAN statements in the usual PLANIT installation process and converts them to equivalent TACPOL statements.

It should be noted that the term, FORTRAN-to-TACPOL Translator is not being used because that has the connotation of a much more general program than the one needed for this particular task. Called TRANSL, this translator program is designed explicitly to translate the programs associated with the PLANIT system and particularly the PLANIT system FORTRAN statements.

Portability of the PLANIT system was achieved by writing the system code in a unique, machine-independent higher level language. Although this language bears some similarity to FORTRAN, it is necessary that each of the several thousand statements be preprocessed by a system generation program before the result is submitted to a FORTRAN compiler. In system generation, machine and configuration parameters are introduced into the code in such a way that the resulting FORTRAN statements will then be suitable for the intended machine. Therefore TRANSL will only be translating FORTRAN statements of the type that the PLANIT system generator produces. This reduces the magnitude of the translation task significantly and makes possible the guarantee of 100% success in the translation of these FORTRAN statements into TACPOL.

The PLANIT System Generator program is written in the FORTRAN IV language in such a way that it, too, can be installed easily anywhere that a FORTRAN IV compiler exists. Because of adherence to strict programming standards, the Generator can also be translated into TACPOL by using the TRANSL program. Expected translation failures will occur in four types of statements: READ, WRITE, FORMAT and END FILE. These statements are few and are conveniently clustered at the beginning of the program for ease in changing by hand.

The installation process for PLANIT and the PLANIT System Generator is documented in detail elsewhere. Figure 1 shows the general sequence to be followed in the normal installation of PLANIT. Note that the production of FORTRAN is simply an intermediate step in the installation process and has no further function after the program has been compiled. By the time PLANIT is running, all connections to FORTRAN are gone. Figure 2 shows the TACPOL translation to be simply an inserted step in the installation sequence. Note that there is no change at the top, i.e. the manner in which the system generation is done. Also note that, like FORTRAN, all connections to TACPOL disappear when PLANIT is ready to run. Note, too, that both installation methods require the writing of a similar locally designed interface program (MIOP). Therefore, the use of the TRANSL program is just another step in the same general process.

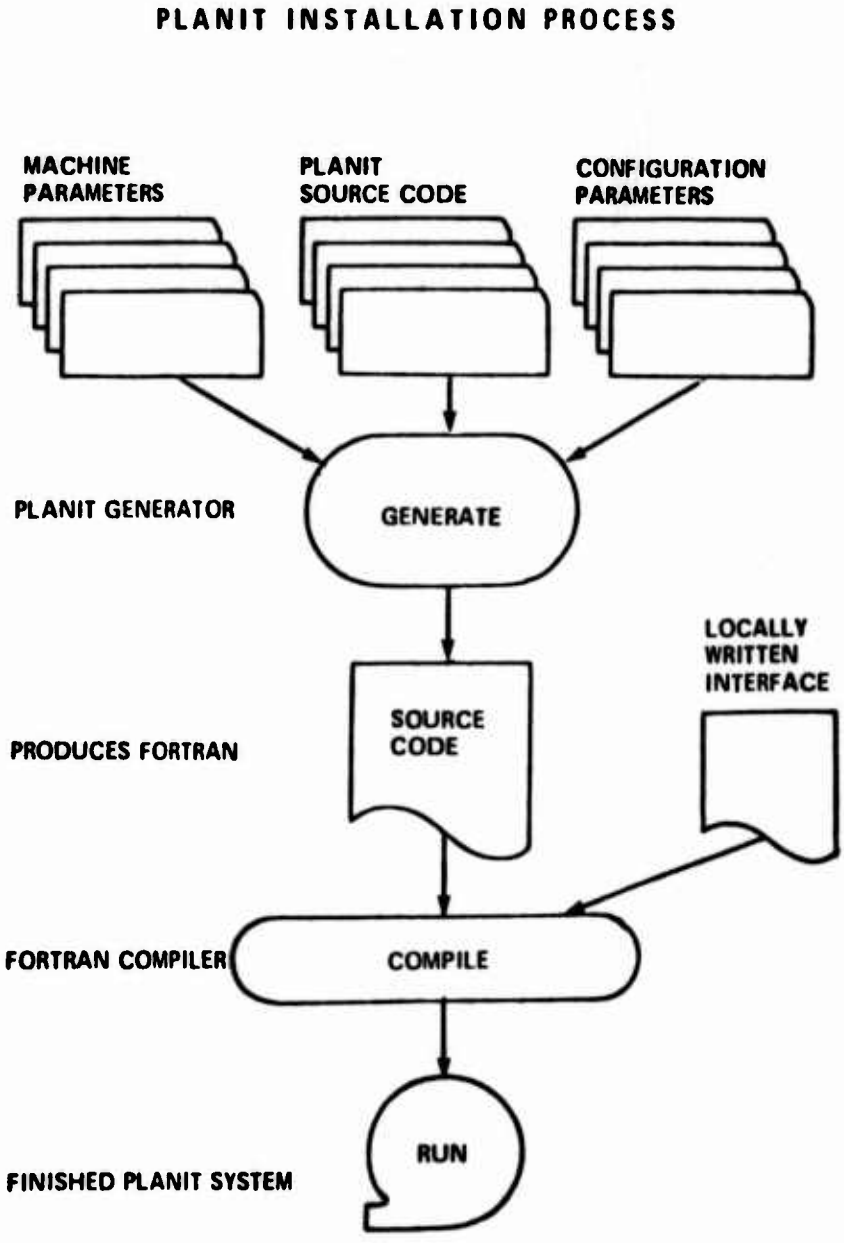

Figure 1. A typical sequence of steps for installing the PLANIT system on a commercial computer. The three top files plus the PLANIT Generator are furnished. Installers modify the parameters to fit local needs prior to system generation. A local FORTRAN IV compiler is required.

PLANIT INSTALLATION PROCESS

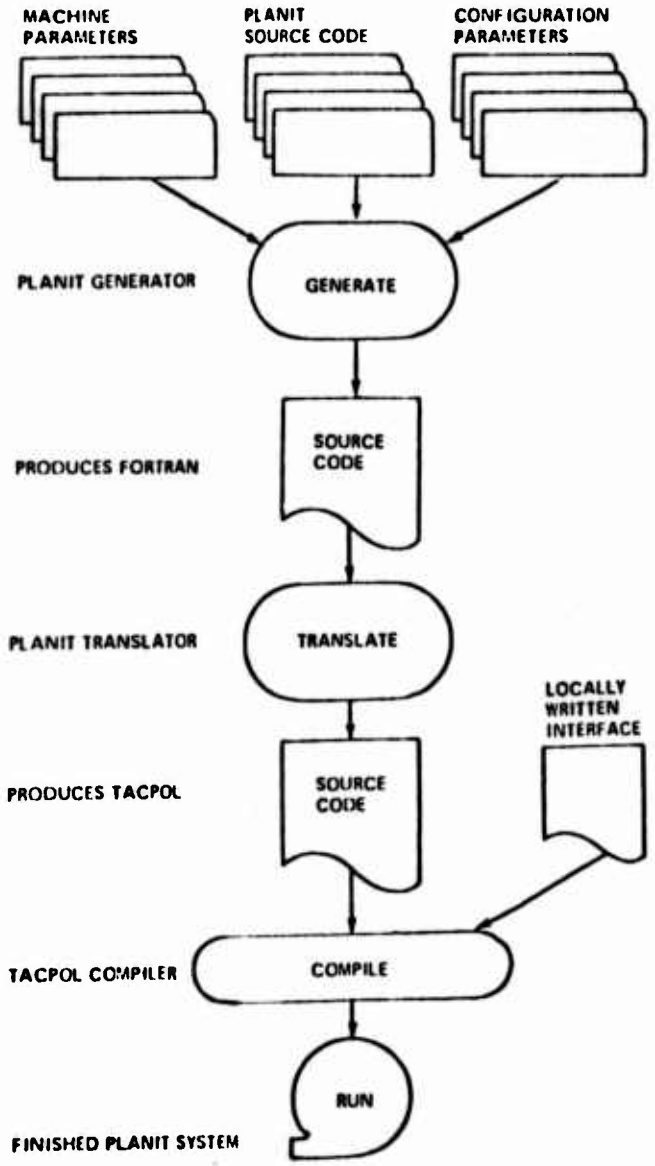

Figure 2. The sequence of steps for installing PLANIT on the ANGYK-12 computer. The three top files plus the PLANIT Generator and the PLANIT Translator are furnished. Parameters have already been set appropriately. A local TACPOL compiler is required.

Finally, for convenience, TRANSL was coded in FORTRAN IV according to the same standards which were followed in the coding of the PLANIT System Generator program. This makes the TRANSL program, like the Generator program, completely portable even though it was written specifically for the ANGYK-12 implementation of PLANIT. The coding of TRANSL in this fashion has two advantages: 1) it could be modified for use in another specialized application at a considerable saving, and 2) the procedures for installing and running TRANSL are nearly identical to those for the Generator. Thus, for the reader who is already familiar with procedures for installing the PLANIT System Generator, the only differences will be that TRANSL uses only one input file instead of two and writes to one output file but not to the line printer (as does the Generator program). Otherwise, the commented section in the listing of TRANSL which describes the few changes which may be necessary is very similar to the Generator and the requirements for the LDBYTE and SBYTE subroutines are identical for both programs (as well as for PLANIT itself). Thus TRANSL can be used to translate itself into TACPOL if such a version is desirable.

INSTALLATION OF TRANSL

The TRANSL program will require one input file and one output file. Both files will be 80-column card images on disk, tape, or whichever physical device is desired. The device assignment is made in the usual FORTRAN manner. The input file is assigned the FORTRAN logical file number one and the output file, number two. These assignments are made in a FORTRAN READ and WRITE statement, respectively, near the top of the program. There is only one of each.

The input file will contain card images of FORTRAN statements and the output file will contain the equivalent TACPOL statements. Diagnostics, if any, will be inserted appropriately in the output file and a count of such diagnostics (if greater than zero) will be added to the end of the file. The content and format of the diagnostics will be discussed later but the user will probably want to plan on obtaining a listing of the output file before going to the next step during actual installation.

The TRANSL program listing begins with several lines of comments which summarize the information presented here. Especially important in these comments are the few lines enclosed between two rows of asterisks (******). Between these lines are grouped all of the FORTRAN statements which should normally be examined during the installation of the TRANSL program.

The first statement to be considered is "IBYTE=4". This statement signifies that there are four bytes (characters) to the computer word in the design of the machine on which this program is being run. That number is correct for the ANGYK-12 computer as well as for many others. If it is incorrect it must be changed.

If there is a need to open either of the files, that should be done just before or just after the IBYTE=4 statement.

The "END FILE 2" statement causes the writing of the end-of-file mark on the output file and it is closed. This statement may need changing to conform to local requirements. File 1 may also be closed at this point if necessary.

The READ and WRITE statements are the last to be checked in this section. The card image is passed through a common buffer, INBUFF. Both the READ and WRITE statements operate with implicit DO-loops to specify exactly one card of 80 columns at a time. The FORMAT card at label 1000 (above) must agree with the READ and WRITE statements

such that exactly one card image will be read (or written) at a time.

In rare instances a FORTRAN compiler may require the dimension of INBUFF to exactly fit one card. If so, then the dimension of the array INQUE must also be adjusted to the new number and a statement below the asterisk line which now reads "IBFDIM=30" must be changed so that the new dimension number replaces the 30. With these changes, the implicit DO-loops could also be eliminated if the compiler objects to them.

In summary, the changes to be expected are in statements which are located between the two lines of asterisks, and these changes will probably be limited to the two statements, "IBYTE=4" and "1000 FORMAT(20A4)" where the number of bytes per computer word differ from four. If the bytes per word equal four, the Translator program should run just as it is.

The final installation requirement of the TRANSL program is the writing of the LDBYTE and SBYTE subroutines. These two subroutines are called throughout TRANSL as well as throughout the Generator and PLANIT. The calls are the same in all cases--"CALL LDBYTE" or "CALL SBYTE"--with no argument string in the calling sequence. The parameters for doing the prescribed work are taken from COMMON in the manner described below. Note that the description will be the same for these subroutines used by the Generator and PLANIT.

Subroutines LDBYTE and SBYTE both use the same set of three parameters, called IBYT1, IBYT2 and INDEX. The first entry in COMMON shows the locations of the three parameters. Suppose an array, IO, was equivalenced to COMMON. Then IO(IO(1)-3) would be the same as IBYT1, IO(IO(1)-2) would be the same as IBYT2, and IO(IO(1)-1) as INDEX. In other words, the first entry in COMMON contains the "index" of the COMMON location just beyond INDEX. Thus, LDBYTE and SBYTE must use the same COMMON as TRANSL, and the first entry in COMMON will locate the three parameters.

The interpretation of each of the three parameters is as follows:

IBYT1 - The byte number of the first character in a string of contiguously packed characters in COMMON.

IBYT2 - The byte number of the last character in the string described above.

INDEX - The COMMON "index" number to which (or from which) the characters will be unpacked (packed)

Figure 3 illustrates the operation of LDBYTE and SBYTE with the byte numbering and indexing scheme. The bytes (character positions) in COMMON assume sequential numbers such that if COMMON was completely filled with one long string of characters tightly packed with no gaps, then the first character would be in position one (leftmost byte of the first word in COMMON) and the numbers would increase sequentially throughout COMMON. The index, on the other hand, refers to the word number of COMMON where the first word has the number one assigned to it. If COMMON was treated as one long array where each word was an entry in the array, then the index would be equivalent to the entry number, beginning at one.

The subroutine LDBYTE is to unpack characters beginning at byte number IBYT1 through byte number IBYT2, placing a copy of the unpacked characters, one byte per word, back into COMMON beginning at INDEX. The unpacked characters are to be right-justified in the word with leading binary zeros. The only COMMON locations which are to be changed (including the parameters themselves) are the words which receive the unpacked characters.

The subroutine SBYTE does the reverse operation from LDBYTE. It must make a string of the rightmost character of each successive word entry beginning at INDEX and pack that string into COMMON beginning at IBYT1 byte position and ending at IBYT2 byte position without changing byte locations on either end of the newly packed string or elsewhere in COMMON.

Figure 3 assumes a computer with four bytes per word and shows an example where:

IBYT1=22
IBYT2=27
INDEX=41

The repositioning of the copied characters which each subroutine must perform is indicated by the arrows.

The LDBYTE and SBYTE subroutines may be written in any form that is most convenient on the target computer. However, if the same ones are to be used for PLANIT, then care should be taken to insure fast and efficient operation.

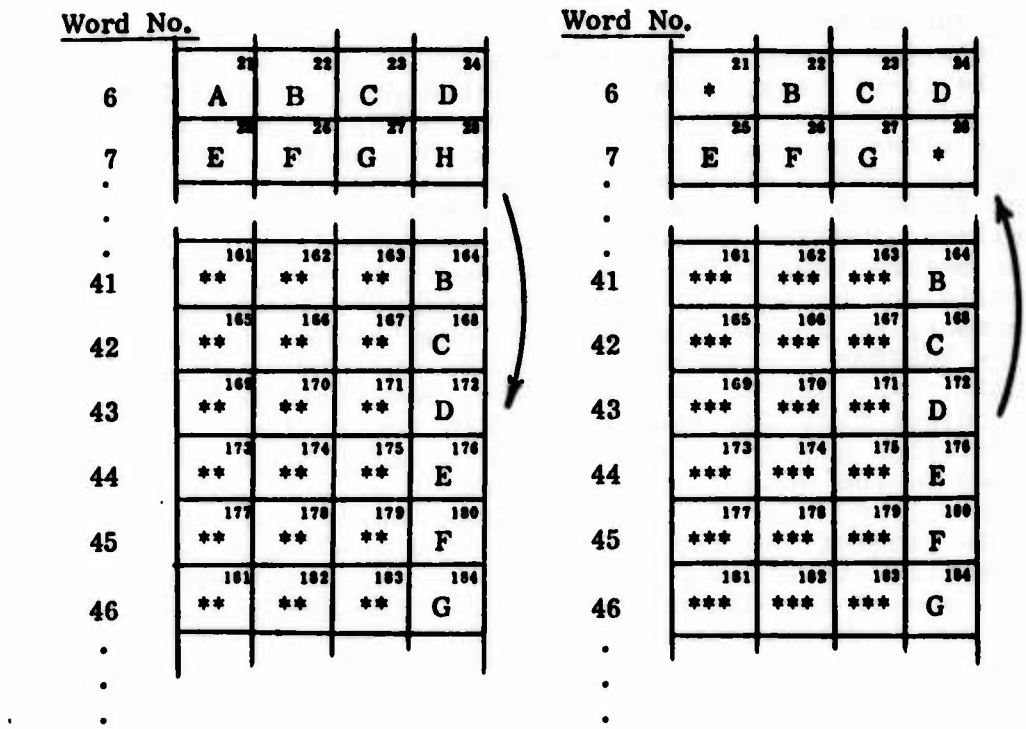

Figure 3. Example of the operations of LDBYTE and SBYTE, both using the same parameter sequence: 22, 27, and 41. Byte numbers are shown in the upper right-hand corner of each cell. One asterisk (*) signifies that the previous value is unchanged, two asterisks (**) designate binary zero (not necessarily the character code for zero), and three asterisks (***) designate values which have no effect whatever on the operation.

APPLICATION OF TRANSL

The TRANSL program accepts FORTRAN IV statements, one at a time, and produces TACPOL statements as required. The TACPOL is suitable for the PSS-B compiler.

The TRANSL program actually performs two separate translation tasks. Each task is run as a one-pass job. In the first task, TRANSL translates only the FORTRAN code statements, passing over the FORTRAN COMMON statements after noting them for proper placement. Other than for COMMON statements, this task produces one or more TACPOL statements for each FORTRAN statement. Continued FORTRAN statements are assembled into a single line before translation, and excessively long TACPOL results are broken into as many continued lines as are necessary as a last step in the translation process. A few FORTRAN statements (including arithmetic IF, CALL LDBYTE, CALL SBYTE, and CALL MIOP) result in more than one output line for the one FORTRAN input line. If the translation attempt fails, the original FORTRAN line is copied to the output file followed by a full line of slashes (i.e. ////////////). The slashes are added as an aid to finding the faulty line in the listing. No further explanation is given of the failure. The problem must either be in one of the documented types of FORTRAN statements that will not translate (i.e. READ, WRITE, FORMAT, END FILE) or the occurrence of incorrect FORTRAN or that which does not conform to PLANIT standards. An error of the latter type might be expected to occur if TRANSL is used to translate a FORTRAN program which is not part of the PLANIT package.

The second task which TRANSL is designed to perform is the creation of TACPOL declare (DCL) statements which will be used by the Compool Generator. These two tasks are separate because a new Compool generation will not be needed for every new translation; only for the first one and each thereafter where any data declarations change. In performing the second task, the TRANSL program must scan all of the FORTRAN code for each initial reference to an item name, and construct a TACPOL data declaration statement for that name. The names may be found in COMMON statements or in the program statements since FORTRAN does not require declaration of simple item names prior to their use. However, TRANSL will not find the needed information in FORTRAN DIMENSION or EQUIVALENCE statements since those are not used in PLANIT coding.

In respect to this second task, the production of the DCL statements, the FORTRAN from the PLANIT system code

has requirements which differ from the FORTRAN from the PLANIT Generator code. The former has the requirement that all data be declared in COMMON; the latter does not have that requirement. A parameter is set for the TRANSL program to indicate which of the two cases it is processing. If it is processing PLANIT system code and encounters an item name which was not previously declared in a COMMON statement, it will insert a warning statement in the output file which says, UNDECLARED ITEM NAME ON THE NEXT LINE////////////// (the slashes mark the line as before). This is simply a check on the PLANIT system code to make sure that all data items are declared as they are supposed to be. However, if it is processing PLANIT Generator code, instead of the warning upon encountering such an undefined item name, TRANSL will simply output an appropriate DCL statement. Note that the FORTRAN for the TRANSL source code is like the PLANIT Generator code and would be translated in the same manner.

One additional manipulation must be done with a few of the data items. Each compiler language has its own list of reserved words. Of course those being used by the PLANIT programs do not conflict with any on the FORTRAN reserved word list. However, some did conflict with the TACPOL reserved word list, namely "VALUE" and "L" which therefore the TRANSL program converts to "XALUE" and "H", respectively, wherever they are encountered.

The remainder of the work of the TRANSL program can be understood most easily in a table which shows the translation rules for each FORTRAN form. The table is shown in Appendix A. Also included is a list of special operation characters for the TRANSL program, all of which occur in column one of the input line.

OPERATION OF TRANSL

In general, TRANSL is a batch program with one input file and one output file. When the program is started, it runs to completion without interruption.

The input file of FORTRAN statements must have 88 cards (lines) inserted prior to the first FORTRAN statement and one card appended to the end. The 88 prefixed cards constitute an initialization deck for the TRANSL program and the information on all but the second card is fixed and not subject to change. Comment cards may be added in addition to the 88 if desired so long as the letter C occurs in the first column. Otherwise, the initialization cards are of four general types, identifiable by the character in column one: (1) the character set card (heads the initialization deck), (2) the operation code card is second in order, having a digit in column one, (3) followed by 85 initialization cards each of which have a blank in columns one and two (with an intermix of comments permissible), (4) ended by a card with a $ character in column one.

The single card which must be added to the end of the FORTRAN statements has a $ character in column one.

Appendix B shows the 88-card initialization deck. Since only the second card contains variable data, it will be the only one that gets further description. Called the operation code, this second card designates which of the four tasks described in the previous section that the TRANSL program is currently expected to execute. The parameter is a single digit in column one having the value, one through four (i.e. 1, 2, 3 or 4). Columns two through 80 on this card are disregarded and may be used as a comment field. The four tasks which are designated will be denoted, Task 1, Task 2, Task 3, and Task 4 corresponding to the parameter settings of one, two, three and four, respectively. Each of the tasks is appropriate for a specific translation need with respect to one or more of the programs in the PLANIT package. These will be described by task number.

Task 1. Task 1 translates PLANIT system FORTRAN code into TACPOL statements. Note that the usual FORTRAN code as it is normally output from the PLANIT Generator program does not have the necessary initialization cards for the TRANSL program nor the ending $ card. However, the PLANIT Generator program does have the facility which allows these cards to be included in the Merge File at the time of system generation. By including the necessary cards in the Merge File, the output from the Generator program is immediately ready to be accepted by the TRANSL program, making a continuous batchstream operation.

A sample Merge File is shown in Appendix C. Notice that the required TRANSL initialization cards are all present but are each shifted to the right five columns to make room for the characters, "*INS " which serves as a directive to the Generator program to insert the cards appropriately. Appendix C is called a sample because some of the parameter cards may change according to the desires of the installer, however the information on the cards which correspond to Appendix B, the initialization deck for the TRANSL program, will remain static. There will be no attempt to explain the parameter cards in Appendix C in this document since that is part of the normal PLANIT installation effort which is documented elsewhere.

The sample Merge File in Appendix C also shows some special &ENDPROG cards which are inserted at the end of each PLANIT subprogram, to be used by the TACPOL compiler in separating the overlays into appropriate subprograms. This will be understood more completely in the TACPOL compiler documentation.

For the execution of Task 1, it is sufficient to say that given a proper Merge File structure prior to the system generation of the FORTRAN, the result is ready to input to the TRANSL program and, in turn, its output is ready for the TACPOL compiler. If no mistakes have been made in the process, 100 percent translation can be expected. However, it may be wise to list the TACPOL at this point to confirm the successful translation. If any errors are flagged, the bad lines can usually be fixed without redoing the process. However, if undefined item names happen to be flagged, then the trouble could point to an error in the original PLANIT code.

Task 2. Task 2 is used to translate the PLANIT Generator program from FORTRAN into TACPOL. It will similarly translate its own FORTRAN source code into TACPOL. The translation of each of these FORTRAN programs should only be a onetime affair as opposed to repeated usage for Task 1. Therefore, a certain amount of post-translation hand repairs is tolerable. The expected repairs are of two kinds: translation failures and rearrangement of code. These are quite well-defined, as follows:

Translation will fail on four kinds of FORTRAN statements, READ, WRITE, FORMAT and END FILE. The number of these which will be encountered in the translation will be small and each will be flagged. The compiler personnel who were responsible for the TACPOL compiler felt that it would be preferable to make the necessary repairs after translation than to attempt to make TRANSL do the work. There will also probably be a need to add appropriate OPEN

calls since FORTRAN does this implicitly. The place where these should be added is documented in comments in the program listing.

The second expected failure is in the placement of subroutines with respect to the main program. This will be no problem for the TRANSL program because it has no subroutines. However, the PLANIT Generator program has two and, like most FORTRAN programs, they follow the main program. In TACPOL programs, procedures (i.e. TACPOL equivalent of subroutines) must precede the main program. Therefore, the translation of the PLANIT Generator code must be rearranged, placing the two procedures (subroutines) ahead of the first main program statement and, since the first subroutine also calls the second, the order of these must also be reversed. Therefore, with respect to the arrangement of the main program and the two subroutines in FORTRAN, the TACPOL version must be rearranged, placing the last procedure (subroutine) first, then working backward to the next procedure, and finally, the main program. Again, making the TRANSL program do this automatically would not be worthwhile, especially in view of how little it would get used.

Note that the initialization deck for Task 2 is identical for all tasks except for the change of the digit in column one of the second card from a "1" to a "2" (or "3" or "4" in the case of the other tasks). Also note that no batchstream is involved as in Task 1 since the code being translated is already FORTRAN and is not generated. Therefore, the 88-card deck as is shown in Appendix B (with the second card changed) is to be appended to the FORTRAN program ahead of the first FORTRAN card, and a terminal $ card ($ in column one) is to be appended to the end. The program is then ready to be input to the translator. The output should be listed, the translation failures (READ, WRITE, FORMAT and END FILE) fixed, OPEN calls added if necessary, code rearranged if necessary, and the DCL statements from Task 3 incorporated appropriately. The result is then ready for the TACPOL compiler.

Task 3. The purpose of Task 3 is to produce data declaration (DCL) statements to go with the TACPOL code which was produced under Task 2. The input file for Task 3 is set up identical to that for Task 2 except for the task number change on the second card. The result will be a set of DCL statements headed by a TACPOL PROC card. The PROC card will be of the form, PLANIT: PROC; where PLANIT is the program name. That name may be changed if desired. In the case of translated code for the translator program,

since it contains no subroutines, the entire set of DCL statements and the PROC card should be added just as they are to the front of the TACPOL code. If the translation failures have already been fixed, that program will be ready to compile.

The presence of subroutines in the PLANIT Generator program code makes the proper placement of the DCL statements more difficult. In general, the DCL statements which correspond to the COMMON items, together with the DCL statements for those items which are unique to the main program should be placed at the head of the main program. Use the PROC card here, too, as the first card of the main program. The subroutines each get only those DCL statements which are unique to the subroutines; they do not receive a copy of the items which were made up from COMMON statements. Again, the DCL statements immediately follow the PROC card. Two procedures will be explained for making the proper placement of the DCL statements, mainly in order to convey the intent. Either procedure may be followed.

Procedure one would be to execute all of Task 3 on the entire input file as described above. The resulting DCL statements would then have to be sorted and placed as follows: Mark the output lines in the DCL statements where they start repeating due to the encounter of a new set of COMMON statements. All statements from the PROC card to the first mark, inclusive, are to be placed at the head of the main program. Delete the next set of cards after the first mark including all that correspond to the COMMON statements which were translated. Place only the remaining DCL statements to the next mark after the PROC card in the subroutine (procedure) from which they came. The statements which corresponded to COMMON are omitted because TACPOL uses them globally from the main program if the DCL statement is not repeated in the procedure. Otherwise, item separation will occur. Repeat the process for the remaining set of DCL statements.

Alternately, the TRANSL program could be run three times in Task 3 for the Generator source code, first separating the FORTRAN into main, and each of the two subroutines. Run TRANSL for the last subroutine as if it were an entire program. Delete the DCL statements that correspond to the COMMON statements and place the remaining DCL statements ahead of the corresponding TACPOL procedure (after the PROC card). Repeat the process for the next subroutine. For the main program, keep all of the DCL statements including those from COMMON and head them with an appropriate PROC card (which will become the program name) and place them at the head of the TACPOL main code section.

The source of the confusion lies in the fact that FORTRAN expects its COMMON data to be listed repeatedly, a copy in the main program and each of the subroutines. Local data is often not declared. TACPOL, on the other hand, implements the COMMON data concept in procedures (subroutines) by omitting the declaration and thereby using the declaration in the calling code by default, calling it a globally defined item. Items which are declared in the procedure are strictly local items to that procedure.

FORTRAN programmers will recognize that in COMMON statements, the exact list of item names need not repeat. However, one of the constraints for the coding of PLANIT-related programs is that all COMMON lists within a program be identical and that no item is explicitly declared in any place other than COMMON (i.e. there is no use of DIMENSION, EQUIVALANCE, TYPE, etc.). To deviate from these conventions would confuse the translator program which only reinforces the conclusion that TRANSL is not a general purpose FORTRAN-to-TACPOL translator.

Task 4. Task 4 generates the DCL statements for the PLANIT system TACPOL code generated in Task 1. The essential difference between Task 3 and Task 4 is in the constraint that all items used by the PLANIT code must be explicitly defined (declared) in the COMMON list. Those found in the COMMON list will be translated into TACPOL DCL statements. Any found in the code which were not previously declared in the COMMON list will (unlike Task 3) produce an error in the output file. To run Task 4, the input file from Task 1 can be changed such that the second card shows a "4" instead of a "1" and the TRANSL job be run again. This time the result should be DCL statements. Since each overlay (subprogram) will have its own COMMON list, several DCL lists will result only one of which will be useful.

Task 4 can be shortened considerably. Since all error checking was already done in Task 1, the input file for Task 4 can be shortened to any point beyond the end of the first listing of COMMON. Therefore, insert a "$" card ("$" in column one) after the last COMMON statement of the first COMMON listing in the input file and TRANSL will stop when the $ is encountered. The result will be a single list of DCL statements, corresponding line-by-line with the COMMON statements. This is all that is needed. Note that this file into which the "$" card is being inserted is the output file from the PLANIT Generator program. The Generator Merge file is set up in such a way so that the output file is right for input

to the TRANSL program. It is the second card of this file that gets changed from a "1" to a "4" and also a "$" card must be inserted beyond the end of the first listing of COMMON. It might be helpful to note that the placement of the "$" card is not critical so long as it follows the COMMON cards. Therefore it could be arbitrarily inserted at, say, line 600 which is well beyond the COMMON cards, perhaps saving the listing of the file to find the exact place.

Having generated a set of DCL statements which correspond line-for-line with any of the several identical COMMON lists, the output file (of DCL cards) will then be properly formatted and ready to be inserted into the procedure which generates the COMPOOL for the TACPOL version of the PLANIT system. The operation of the COMPOOL generator is described elsewhere, being a part of the standard ANGYK-12 software facility.

FINAL INSTALLATION CONSIDERATIONS

These are some observations that seemed best to be presented near the end, when the reader has more of the total plan in view.

PROC Cards. It was noted that while both Task 3 and Task 4 produce DCL statements, only Task 3 generates a PROC card to head the statements (in the form, PLANIT: PROC;). This is because the PROC card is needed in that position to head the main program of the kind that is translated in Tasks 2 & 3, namely, the PLANIT Generator and the Translator. However, it may be desirable to change the procedure names to something more appropriate like: GENER: PROC; or TRANSL: PROC;. The PROC card is not generated in Task 4 because these DCL statements are destined for the COMPOOL generator. In this case, the needed PROC card is supplied through the Merge File prior to generation and is placed appropriately during translation in Task 1.

Special Subroutine (Procedure) Calls. The TRANSL program makes the appropriate translation for FORTRAN subroutines and the calls to them provided that no argument strings exist in the calling sequences. It is noted here for information that the call format produced by the TRANSL program will be considerably different in Task 1 than in Task 2. In Task 1, what amount to subroutine calls actually invoke a level change, initiating a new program. This is peculiar to the special PLANIT operating system which was developed for this computer. In Task 2, the calling format is standard TACPOL. The cause of special interest is the calling sequence for the special procedures, LDBYTE and SBYTE. In the PLANIT system, as translated by Task 1, these are separate programs operating at their own level. However, for the PLANIT Generator and the Translator programs, both of which use LDBYTE and SBYTE routines which are identical to those for PLANIT, they are placed at the head of the respective programs in normal TACPOL fashion and are invoked through the standard calling format which is produced under Task 2. Therefore, the TACPOL versions of the PLANIT Generator and the Translator programs should each begin with these two TACPOL procedures, LDBYTE and SBYTE.

In summary, the final procedure arrangement for the TACPOL version of the PLANIT Generator program should be (from first to last):

1. LDBYTE
2. SBYTE
3. NUMBER (Last FORTRAN subroutine)
4. BRCKET (First FORTRAN subroutine)
5. Main program

The Main program should start with its PROC statement, followed by the DCL statements corresponding to FORTRAN COMMON and then the DCL statements for the items local to the Main program. Each of the procedures should begin with appropriate PROC statements, followed by only those DCL statements that are for items local to that procedure.

The final arrangement for the TACPOL version of the Translator program is comparable, as follows:

1. LDBYTE
2. SBYTE
3. Main program

The arrangement of the PROC cards and DCL statements is the same as above.

APPENDIX A

Table of rules by which the TRANSL program translates FORTRAN particles into corresponding TACPOL particles.

	FORTRAN Particle (becomes)	TACPOL Particle	Comment
1.	Label (e.g. 1)	Q1:	FORTRAN labels begin in column 1, TACPOL in column 2.
2.	CONTINUE	(deleted)	
3.	DO (e.g. DO 1 I=1,10)	DO I=1 BY 1 TO 10	An END; will be placed appropriately after label 1.
4.	DO (e.g. DO 2 I=J,K,3)	DO I=J BY 3 TO K	An END; will be placed appropriately after label 2.
5.	IF (e.g. IF(···) expression)	IF(···) THEN expression;	

6.	IF (e.g. IF(···) 1,2,3	IF(··· LT 0) THEN GOTO Q1;
		IF(... EQ 0) THEN GOTO Q2;
		GOTO Q3;
			Note: one of the three statements will be omitted if its corresponding label appears on the next line.

7.	COMMON		Only noted for data check. Not translated to output.
8.	First occurrence of an integer item name (e.g. I)	DCL I BIN FIXED;	
9.	First occurrence of a real item name (e.g. X)	DCL X BIN FIXED (62,15);	
10.	CALL (e.g. CALL BRCKET)	CALL BRCKET;	
11.	CALL LDBYTE CALL SBYTE CALL MIOP		In Task 1, these three calls are replaced by direct code which causes an appropriate program level change in accordance with the PLANIT Operating System which was developed for the ANGYK-12.
12.	SUBROUTINE (e.g. SUBROUTINE PLAN1)	PLAN1: PROC;	Use of argument strings is invalid.

	FORTRAN Particle	(becomes)	TACPOL Particle	Comment
13.	FORMAT			Flagged and not translated.
14.	READ			Flagged and not translated.
15.	WRITE			Flagged and not translated.
16.	PAUSE		GOTO QREND;	The label QREND: will be added just before the END; statement.
17.	STOP		(deleted)	STOP can only occur just prior to the END statement.
18.	END		END;	
19.	ENDFILE			Flagged and not translated.
20.	GO TO (e.g. GO TO 1)		GOTO Q1;	
21.	GOTO (e.g. GOTO 1)		GOTO Q1;	
22.	ABS(···)		ABS(···)	
23.	IABS(···)		ABS(···)	
24.	ALOG(···)		LONG(LN(SHORT(···)))	
25.	SQRT(···)		LONG(SQRT(SHORT(···)))	
26.	SIN(···)		SINFUN(···)	The SIN and COS calls are translated to calls on special SINFUN and COSFUN procedures which convert the radian arguments into "BAMS" for the TACPOL compiler. The SINFUN and COSFUN procedures are inserted in the Generator jobstream appropriately.
27.	COS(···)		COSFUN(···)	

	FORTRAN Particle (becomes)	TACPOL Particle	Comment
28.	RETURN	GOTO QREND;	
29.	.NE. .LT. .LE. .EQ. .GE. .GT.	NE LT LE EQ GE GT	
30.	.AND. .OR.	)AND( )OR(	
31.	=-IABS	=-1*ABS	Correction for compiler error.
32.	=-ABS	=-1.0L*ABS	Correction for compiler error.
33.	Real division, e.g. X/Y or (···)/Y	TRUNC(···,30)/Y	Correct for shifting of decimal places.
34.	Computed GOTO, e.g. GO TO(1,2,3),I	CALL SWITCH(I,Q1,Q2,Q3);	
35.	Mode conversion, e.g. X=I I=X	X=LONG(I) I=SHORT(X)	The arguments of LONG and SHORT can be arbitrary expressions.
36.	Real literal numeric conversion, e.g. 2. 2.0 2.1	 2.0L 2.0L 2.1L	
37.	Reserved word conversion: VALUE L	 XALUE H	VALUE and L have special meaning in TACPOL.

	FORTRAN Particle (becomes)	TACPOL Particle	Comment
38.	Comments: the line beginning with "C" in column one.	The characters /* go in columns 2 & 3 (column 1 blanked) and the characters */ go in columns 71 & 72.	
39.	The letter "B" in column one.	Blank column one and output the line with no further modification.	
40.	Column one not a digit, blank, "B" or "C"	Output the line as is.	
41.	Column one a digit or blank.	FORTRAN line to be translated according to the above rules.	
42.	Complete FORTRAN line.	Append a semicolon (;) to the end of the translated line.	
43.	Characters **	Replace entire line with NDKTR=18;	TACPOL does not accept variable exponents.
44.	$$ (end of job card)	&ENDPROG	TACPOL end of job card.

APPENDIX B

Listing of the initialization cards which must be inserted in the input file so that they precede the first FORTRAN statement to be translated.

```
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ+-*/().?=,:;'@ #] ?$    00000100
1 -- NTYPE PARAMETER (COL. 1 ONLY). DESCRIPTION FOLLOWS:    00000200
  CONTINUE                                                  00000300
  DO                                                        00000400
  IF                                                        00000500
  COMMON                                                    00000600
  CALL                                                      00000700
  SUBROUTINE                                                00000800
  FORMAT                                                    00000900
  READ                                                      00001000
  WRITE                                                     00001100
  PAUSE                                                     00001200
  STOP                                                      00001300
  PROC                                                      00001400
  END                                                       00001500
  ENDFILE                                                   00001600
  GO                                                        00001700
  GOTO                                                      00001800
  ABS                                                       00001900
  IABS                                                      00002000
  ALOG                                                      00002100
  SQRT                                                      00002200
  SIN                                                       00002300
  COS                                                       00002400
  RETURN                                                    00002500
  N/U                                                       00002600
  N/U                                                       00002700
  VALUE                                                     00002800
  L                                                         00002900
  N/U                                                       00003000
  N/U                                                       00003100
  N/U                                                       00003200
  .NE                                                       00003300
  .LT                                                       00003400
  .LE                                                       00003500
  .EQ                                                       00003600
  .GE                                                       00003700
  .GT                                                       00003800
  .OR                                                       00003900
  .AND                                                      00004000
  N/U                                                       00004100
  MSGTBL OR MSGDR NOT LARGE ENOUGH.                         00004200
  LT                                                        00004300
  EQ                                                        00004400
  GT                                                        00004500
   PY 1 TO                                                  00004600
  ; END                                                     00004700
  THENX                                                     00004800
   XX 0                                                     00004900
   XX C.CL                                                  00005000
   C                                                        00005100
  CALL SWITCH(                                              00005200
  SHORT(TRUNC(                                              00005300
  LONG(                                                     00005400
  ))                                                        00005500
  )                                                         00005600
  FUN                                                       00005700
  N/U                                                       00005800
  (SHORT                                                    00005900
  LN                                                        00006000
```

```
      CCL                          BIN FIXED;                                    00006100
      CCL                          BIN FIXED (62,15);                            00006200
      CREAC: END                                                                 00006300
      ECTC CREND                                                                 00006400
      UNDECLARED ITEM NAME ON NEXT LINE ////////////////////                     00006500
      TOO MANY ITEM NAME ENTRIES FOR THE DICTIONARY.                             00006600
      TOTAL COUNT OF FLAGGED LINES EQUALS:                                       00006700
      LDBYTE                                                                     00006800
      SBYTE                                                                      00006900
      MIOP                                                                       00007000
      N/L                                                                        00007100
      N/L                                                                        00007200
            TXP =X'1EC5' LDBYTE CALL                                             00007300
            TXP =X'17C5' SBYTE CALL                                              00007400
            TXP =X'15C5' MIOP CALL                                               00007500
      N/L                                                                        00007600
      N/L                                                                        00007700
            CODE:                                                                00007800
            BRT R'C',1                                                           00007900
            END                                                                  00008000
      PLANIT: PROC:                                                              00008100
      &EXECPROG                                                                  00008200
      ,3C)                                                                       00008300
      TRUNC(                                                                     00008400
        NCKTR=18:                                                                00008500
      1,CL*                                                                      00008600
      1*                                                                         00008700
   $$   END OF PARMS FOR TRANSL PROGRAM.                                         00008800
```

APPENDIX C

Sample listing of a PLANIT Merge file with the Translator initialization deck embedded in such a way that the resulting FORTRAN will lead off with the required initialization deck. This constitutes a batchstream for going from the original PLANIT code through FORTRAN to TACPOL with no intervening manipulation.

```
*GENERATOR OPTIONS (COMM=1, MAP=1, STOP=1, PROC=4)                    CCCCC100
BYTWRT=4                                                              CCCCC200
BITWRD=32                                                             CCCCC300
INTPREC=9                                                             CCCCC400
SCIPREC=9                                                             CCCCC500
FLTPNT=2                                                              CCCCC600
ARITHMETIC=2                                                          CCCCC700
FACTLIM=16                                                            CCCCC800
CLCKTYPE=2                                                            CCCCC900
SECCIV=1C                                                             CCCC1000
UPERLIST=1                                                            CCCC1100
UPERRCR=1C                                                            CCCC1200
UPERPLNCH=50                                                          CCCC1300
CRLF=2                                                                CCCC1400
CRTRN=13                                                              CCCC1500
LFEED=1C                                                              CCCC1600
RECSZF=32                                                             CCCC1700
TTYLINF=72                                                            CCCC1800
CHARSEC=150                                                           CCCC1900
PRLNSZ=72                                                             CCCC2000
MAXLINES=99999                                                        CCCC21CC
NUMCH=9                                                               CCCC2200
INBUFF=EC                                                             CCCC2300
OUTBUF=520                                                            CCCC2400
DSKSZF=5C                                                             CCCC2500
VARENT=P                                                              CCCC2600
STACK=1C                                                              CCCC2700
FORMS=2                                                               CCCC28CC
OPERCPT=1                                                             CCCC2900
LINKSZE=10                                                            CCCC3CCC
NMRSPF=2                                                              CCCC3100
CERLC=1                                                               CCCC3200
NOPSZE=6C                                                             CCCC33CC
FUNLVL=9                                                              CCCC3400
FRMTSZE=110                                                           CCCC35CC
LABTSZE=44                                                            CCCC3600
RECFPM=10                                                             CCCC3700
MSESZE=5                                                              CCCC3800
SREC=5CC                                                              CCCC3900
CALCNAMES=35                                                          CCCC4CCC
MATSZE=4CC                                                            CCCC41CC
FUNSZE=2CC                                                            CCCC4200
PAGESKIP=50                                                           CCCC4300
LINESKIP=50                                                           CCCC4400
SPACING=1                                                             CCCC4500
*PLANIT FILE                                                         CCCC46CC
C:                                                                    CCCC47CC
C:ALL LINES BELCW ARE PREFIXED WITH THE CHARACTERS,                   CCCC48CC
C:'*INS ' TO BUILC A JOB STREAM GCING THRCUGH THE GENERATOR.          CCCC49CC
C:                                                                    CCCC5CCC
*INS C12345678SABCDEFGHIJKLMNCPQRSTUVWXYZ+-*/().%=.:;'@ #|_?$        CCCC51CC
*INS 1 -- NTYPE PARAMETER (COL. 1 ONLY). DESCRIPTICN FCLLOWS:         CCCC52CC
C: ITEM NTYPE IS A TRANSLATCR RUN CPTION WITH FCUR LEGAL              CCCC53CC
C: VALUES: (1) PLANIT FORTRAN CODE, (2) CTHER FORTRAN CCCE,           CCCC54CC
C: (3) DCL STATEMENTS FCR OTHER FORTRAN CCDE, (4) DCL                 CCCC55CC
C: STATEMENTS FOR PLANIT.  SEE THE NTYPE COMMENTS IN THE              CCCC5600
C: PRCGRAM LISTING FOR MORE EXPLANATION.                              CCCC57CC
C:                                                                    CCCC58CC
*INS    CONTINUE                                                      CCCC5900
*INS    CO                                                            CCCC6000
```

```
*INS    IF                                                              00000610
*INS    COMMON                                                          00000620
*INS    CALL                                                            00000630
*INS    SUBROUTINE                                                      00000640
*INS    FORMAT                                                          00000650
*INS    READ                                                            00000660
*INS    WRITE                                                           00000670
*INS    PAUSE                                                           00000680
*INS    STOP                                                            00000690
*INS    PROC                                                            00000700
*INS    END                                                             00000710
*INS    ENDFILE                                                         00000720
*INS    GO                                                              00000730
*INS    GOTO                                                            00000740
*INS    ABS                                                             00000750
*INS    IABS                                                            00000760
*INS    ALOG                                                            00000770
*INS    SQRT                                                            00000780
*INS    SIN                                                             00000790
*INS    COS                                                             00000800
*INS    RETURN                                                          00000810
*INS    N/U                                                             00000820
*INS    N/U                                                             00000830
*INS    VALUE                                                           00000840
*INS    L                                                               00000850
*INS    N/U                                                             00000860
*INS    N/U                                                             00000870
*INS    N/U                                                             00000880
*INS    .NE                                                             00000890
*INS    .LT                                                             00000900
*INS    .LE                                                             00000910
*INS    .EQ                                                             00000920
*INS    .GE                                                             00000930
*INS    .GT                                                             00000940
*INS    .OR                                                             00000950
*INS    .AND                                                            00000960
*INS    N/U                                                             00000970
*INS    PSGTBL OR PSGDR NOT LARGE ENOUGH.                               00000980
*INS    LT                                                              00000990
*INS    EQ                                                              00001000
*INS    GT                                                              00001010
*INS     BY 1 TO                                                        00001020
*INS     : END                                                          00001030
*INS    THENX                                                           00001040
*INS     XX 0                                                           00001050
*INS     XX 0.0L                                                        00001060
*INS     0                                                              00001070
*INS    CALL SWITCH(                                                    00001080
*INS    SHORT(TRUNC(                                                    00001090
*INS    LONG(                                                           00001100
*INS    ))                                                              00001110
*INS    )                                                               00001120
*INS    FLN                                                             00001130
*INS    N/U                                                             00001140
*INS    (SHORT                                                          00001150
*INS    LN                                                              00001160
*INS    DCL                    BIN FIXED;                               00001170
*INS    DCL                    BIN FIXED (62,15);                       00001180
*INS    ORENC: ENC                                                      00001190
*INS    GOTO ORENC                                                      00001200
```

```
      *INS    UNDECLARED ITEM NAME ON NEXT LINE ////////////////////                  00012100
      *INS    TOO MANY ITEM NAME ENTRIES FOR THE DICTIONARY.                         00012200
      *INS    TOTAL COUNT OF FLAGGED LINES EQUALS:                                   00012300
      *INS    LCRYTE                                                                 00012400
      *INS    SBYTE                                                                  00012500
      *INS    MIOP                                                                   00012600
      *INS    N/U                                                                    00012700
      *INS    N/U                                                                    00012800
      *INS            TXP =X'1605' LDBYTE CALL                                       00012900
      *INS            TXP =X'17C5' SBYTE CALL                                        00013000
      *INS            TXP =X'1905' MIOP CALL                                         00013100
      *INS    N/U                                                                    00013200
      *INS    N/U                                                                    00013300
      *INS            CODE:                                                          00013400
      *INS            RPT R'C',1                                                     00013500
      *INS            END                                                            00013600
      *INS    PLANIT: PROC:                                                          00013700
      *INS    &ENDPROG                                                               00013800
      *INS    .30)                                                                   00013900
      *INS    TPLNCL                                                                 00014000
      *INS      NDKTR=12:                                                            00014100
      *INS    1.CL*                                                                  00014200
      *INS    1*                                                                     00014300
      *INS $$   END OF PARMS FOR TRANSL PROGRAM.                                     00014400
      C:                                                                            00014500
            SUBROUTINE PLANIT                                                       00014600
      *COMMON                                                                       00014700
      *PLANIT FILE -- INC. DISTRIBUTOR.                                             00014800
      C:PARTITION ASSIGNMENTS FOR PRESET AND CFILE.                                 00014900
            IPART(<PRESET>)=7                                                       00015000
            IPART(<CFILE>)=7                                                        00015100
      *PLANIT                                                                       00015200
            IANPRT=N)IPRI.                                                          00015300
            GO TO(3C1,3C2,303,3C4,305,306,307,308),NXTPRT                           00015400
      3C1   CONTINUE                                                                00015500
      *INS P        CODE:                                                          00015600
      *INS B        RPT R'0',1                                                     00015700
      *INS B        TXP =X'41C7'                                                   00015800
      *INS B        END                                                            00015900
            GO TO 3C1EE                                                           00016000
      302   CONTINUE                                                              00016100
      *INS B        CODE:                                                          00016200
      *INS B        RPT R'C',1                                                     00016300
      *INS B        TXP =X'42C7'                                                   00016400
      *INS B        END                                                            00016500
            GO TO 3C1EE                                                           00016600
      303   CONTINUE                                                              00016700
      *INS B        CODE:                                                          00016800
      *INS B        RPT R'0',1                                                     00016900
      *INS B        TXP =X'43C7'                                                   00017000
      *INS B        END                                                            00017100
            GO TO 3C1EE                                                           00017200
      304   CONTINUE                                                              00017300
      *INS B        CODE:                                                          00017400
      *INS B        RPT R'0',1                                                     00017500
      *INS B        TXP =X'44C7'                                                   00017600
      *INS B        END                                                            00017700
            GO TO 3C1EE                                                           00017800
      3C5   CONTINUE                                                              00017900
      *INS B        CODE;                                                          00018000
```

```
      *INS P          RBT R'0',1                                          C0018100
      *INS B          TXP =X'45C7'                                        C0018200
      *INS B          END                                                 C0018300
            GO TO 3C1EE                                                  C0018400
      30E   CONTINUE                                                     C0018500
      *INS B          CODE:                                               C0018600
      *INS B          RBT R'0',1                                          C0018700
      *INS P          TXP =X'46C7'                                        C0018800
      *INS B          END                                                 C0018900
            GO TO 3C1EE                                                  C0019000
      3C7   CONTINUE                                                     C0019100
      *INS B          CODE:                                               C0019200
      *INS B          RBT R'0',1                                          C0019300
      *INS B          TXP =X'4707'                                        C0019400
      *INS B          END                                                 C0019500
            GO TO 3C1EE                                                  C0019600
      3C8   CONTINUE                                                     C0019700
      *INS B          CODE:                                               C0019800
      *INS B          RBT R'0',1                                          C0019900
      *INS P          TXP =X'48C7'                                        C0020000
      *INS B          END                                                 C0020100
            GO TO 3C1EE                                                  C0020200
      *PLANIT                                                            C0020300
            GO TO(31111,32222,32222,<MICP>,<SCHEDULER>00,<HANG>00        C0020400
                1    .<FORMAT>,<FORFLT>,<FORINT>,<WRITE>,<DISKIO>        C0020500
                2    .<FYND>,32222,<LABSRCH>,<PARSE>,32222,<STRINGR>     C0020600
                3    .<EDIT>,<ALLOCATE>,<TAPEIO>,<LESSRCH>,<FORWRD>,<REED>C0020700
                4    .<BKGRND>),IGOTO                                    C0020800
      *PLANIT FILE -- INC. REMAINDER OF DISTRIBUTOR.                      C0020900
      *DELETE LOBYTE                                                      C0021000
      *DELETE SAVTE                                                       C0021100
      *PLANIT FILE -- INC. MICP                                           C0021200
      *PLANIT FILE -- INC. SCHEDULER                                      C0021300
      *PLANIT FILE -- INC. HANG                                           C0021400
      *PLANIT FILE -- INC. FORMAT                                         C0021500
      *PLANIT FILE -- INC. FORFLT                                         C0021600
      *PLANIT FILE -- INC. FORINT                                         C0021700
      *PLANIT FILE -- INC. WRITE                                          C0021800
      *PLANIT FILE -- INC. DISKIO                                         C0021900
      *PLANIT FILE -- INC. FYND                                           C0022000
      *PLANIT FILE -- INC. LABSRCH                                        C0022100
      *PLANIT FILE -- INC. PARSE                                          C0022200
      *PLANIT FILE -- INC. STRINGR                                        C0022300
      *PLANIT FILE -- INC. EDIT                                           C0022400
      *PLANIT FILE -- INC. ALLOCATE                                       C0022500
      *PLANIT FILE -- INC. TAPEIO                                         C0022600
      *PLANIT FILE -- INC. LESSRCH                                        C0022700
      *PLANIT FILE -- INC. FORWRD                                         C0022800
      *PLANIT FILE -- INC. REED                                           C0022900
      *PLANIT FILE -- INC. BKGRND                                         C0023000
      *END                                                               C0023100
      31111 STOP                                                         C0023200
            END                                                          C0023300
      *INS GENDPROG                                                      C0023400
            SUBROUTINE PLAN1                                             C0023500
      *COMMON                                                            C0023600
      *INS A     SINFUN:  PROC(X) FIXED(62,15);                          C0023700
      *INS B       DCL X BIN FIXED (62,15) VALUE;                        C0023800
      *INS B       DCL Z BIN FIXED (62,15);                              C0023900
      *INS B       DCL SINDATA BIN FIXED (31,30);                        C0024000
```

```
*INS  B          CODE:                                                    00C24100
*INS  B          LDF   X+R'1',6                                           00C24200
*INS  B          LDF   X+2+R'1',7                                         00C24300
*INS  B          SALC  =32,6                                              00C24400
*INS  B          SARC  =32,6                                              00C24500
*INS  B          MPF   RADBAMS-$,7                                        00C24600
*INS  B          SALC  =16,6                                              00C24700
*INS  B          RBT   R'6',1                                             00C24800
*INS  B          XPF   $+2,6                                              00C24900
*INS  B          ADF   CNES-$,6                                           00C25000
*INS  B          SDF   SINDATA+R'1',6                                     00C25100
*INS  B          XFR   $+4                                                00C25200
*INS CNES        GEN   FX'4CC00000'                                       00C25300
*INS RADBAMS     GEN   FD.15$154943831                                    00C25400
*INS  B          END                                                      00C25500
*INS  B      SINDATA = SIN(SINDATA);                                      00C25600
*INS  B          CODE:                                                    00C25700
*INS  B          LDF   SINDATA+R'1',6                                     00C25800
*INS  B          XPF   $+2,6                                              00C25900
*INS  B          ACF   =X,'7FFF',6                                        00C26000
*INS  B          SARC  =47,6                                              00C26100
*INS  B          SDF   Z+R'1',6                                           00C26200
*INS  B          SDF   Z+2+R'1',7                                         00C26300
*INS  B          END                                                      00C26400
*INS  B      RETURN(Z);                                                   00C26500
*INS  B      END;  /* SINFUN PROC */                                      00C26600
*INS  B     COSFUN:  PROC(X) FIXED(62,15);                                00C26700
*INS  B      DCL X BIN FIXED (62,15) VALUE;                               00C26800
*INS  B      DCL Z BIN FIXED (62,15);                                     00C26900
*INS  B      DCL CCSDATA BIN FIXED (31,30);                               00C27000
*INS  B          CODE:                                                    00C27100
*INS  B          LDF   X+R'1',6                                           00C27200
*INS  B          LDF   X+2+R'1',7                                         00C27300
*INS  B          SALC  =32,6                                              00C27400
*INS  B          SARC  =32,6                                              00C27500
*INS  B          MPF   RADBAMS-$,7                                        00C27600
*INS  B          SALC  =16,6                                              00C27700
*INS  B          RBT   R'6',1                                             00C27800
*INS  B          XPF   $+2,6                                              00C27900
*INS  B          ADF   CNES-$,6                                           00C28000
*INS  B          SDF   COSDATA+R'1',6                                     00C28100
*INS  B          XFR   $+4                                                00C28200
*INS CNES        GEN   FX'4CCOCC00'                                       00C28300
*INS RADBAMS     GEN   FD.15$154943831                                    00C28400
*INS  B          END                                                      00C28500
*INS  B      COSDATA = COS(COSDATA);                                      00C28600
*INS  B          CODE:                                                    00C28700
*INS  B          LDF   COSDATA+R'1',6                                     00C28800
*INS  B          XPF   $+2,6                                              00C28900
*INS  B          ACF   =X,'7FFF',6                                        00C29000
*INS  B          SARC  =47,6                                              00C29100
*INS  B          SDF   Z+R'1',6                                           00C29200
*INS  B          SDF   Z+2+R'1',7                                         00C29300
*INS  B          END                                                      00C29400
*INS  B      RETURN(Z);                                                   00C29500
*INS  B      END;  /* COSFUN PROC */                                      00C29600
        IGOTO=IGOTO-<CALC-1>                                              00C29700
        GO TO(<CALC>,1,1,<EVAL>,<GNOFFS>00                                00C29800
     6     ,<COMPGOTO>,<LOGRITHM>                                        00C29900
     7     ,<DEFFUNCTN>,<MATRIXDEFINE>,<MATRIXARITH>,<SET>               00C30000
```

Alpine Business Forms, Inc. ®

```
      E     ,<REPLY>,<PRINTALIGN>),IGOTO                                  CCC3C1CC
      2     IGOTO=0                                                       CCC3C2CC
      1     RETURN                                                        CCC3C3CC
     *PLANIT FILE -- INC. CALC                                            CCC3C4CC
     *PLANIT FILE -- CALC SW.                                             00003C5CC
     *PLANIT FILE -- CALC SW.                                             00003C6CC
     *PLANIT FILE -- CALC SW.                                             C0003C7CC
     *PLANIT FILE -- CALC SW.                                             CCC3C8CC
     *PLANIT FILE -- CALC SW.                                             CCC3C9CC
     *PLANIT FILE -- INC. EVAL                                            CCC31CCC
     *PLANIT FILE -- INC. ONOFFS                                          CCC311CC
     *PLANIT FILE -- INC. COMPGOTO                                        00031200
     *PLANIT FILE -- INC. LOGRITHM                                        00031300
     *PLANIT FILE -- INC. DEFFUNCTN                                       C0031400
     *PLANIT FILE -- INC. MATRIXDEFINE                                    CCC31500
     *PLANIT FILE -- INC. MATRIXARITH                                     CCC31600
     *PLANIT FILE -- INC. SET                                             CCC317CC
     *PLANIT FILE -- INC. REPLY                                           00031800
     *PLANIT FILE -- INC. PRINTALIGN                                      CCC31900
     *ENC                                                                 CCC32000
           END                                                            00032100
     *INS &ENDPROG                                                        CCC32200
           SUBROUTINE PLAN2                                               00032300
     *COMMON                                                              CCC324CC
           IGOTO=IGOTO-<ALLOW-1>                                          CCC32500
           GO TO(<ALLOW>,<PRINT>,<DROP>                                   CCC326CC
          1  ,<PRINTMATRIX>,<RANK>,<INVERT>,<ARRAY>,<COMBIN>              CCC327CC
          2  ,<HELP>,<STATUS>,<CHANGE>),IGOTO                            CCC328CC
      2     IGOTO=0                                                       CCC329CC
      1     RETURN                                                        CCC33CCC
     *PLANIT FILE -- INC. ALLOW                                           CCC331CC
     *PLANIT FILE -- INC. PRINT                                           00033200
     *PLANIT FILE -- INC. DROP                                            CCC33300
     *PLANIT FILE -- INC. PRINTMATRIX                                     00033400
     *PLANIT FILE -- INC. RANK                                            00033500
     *PLANIT FILE -- INC. INVERT                                          C0033600
     *PLANIT FILE -- INC. ARRAY                                           CCC337CC
     *PLANIT FILE -- INC. COMBIN                                          00033800
     *PLANIT FILE -- INC. HELP                                            00033900
     *PLANIT FILE -- INC. STATUS                                          00034000
     *PLANIT FILE -- INC. CHANGE                                          C0034100
     *ENC                                                                 CCC34200
           END                                                            CCC343CC
     *INS &ENDPROG                                                        CCC34400
           SUBROUTINE PLAN3                                               00034500
     *COMMON                                                              00034600
           IGOTO=IGOTO-<EXECUTE-1>                                        CCC34700
           GO TO(<EXECUTE>,1,<MATCH>,<SCAN>,<DECISION>,<DOCOMM>           CCC348CC
          1  ,<RECYCLE>,<FRMEND>,<EXIT>),IGOTO                            CCC349CC
      2     IGOTO=0                                                       CCC35CCC
      1     RETURN                                                        CCC351CC
     *PLANIT FILE -- INC. EXECUTE                                         00035200
     *PLANIT FILE -- INC. MATCH                                           00035300
     *PLANIT FILE -- INC. SCAN                                            C0035400
     *PLANIT FILE -- INC. DECISION                                        00035500
     *PLANIT FILE -- INC. DOCOMM                                          00035600
     *PLANIT FILE -- INC. RECYCLE                                         00035700
     *PLANIT FILE -- INC. FRMEND                                          CCC35BCC
     *PLANIT FILE -- INC. EXIT                                            00035900
     *ENC                                                                 CCC36000
```

```
      END                                                                   C0C3E1CC
*INS &ENDPROG                                                               C0C3E2CC
      SUBROUTINE PLAN4                                                      C0C3E3CC
*COMPCN                                                                     00036400
      IGOTO=IGOTC-<STUSRCH-1>                                               00036500
      GO TO(<STUSRCH>,<LOGIN>,<ADDEL>,<ATTACH>,<SAVE>                       00036600
     1   ,<LOCKER>,<RESTART>,<FRMERE>),IGOTC                                00036700
2     IGOTO=0                                                               00036800
1     RETURN                                                                00036900
*PLANIT FILE -- INC. STUSRCH                                                00037000
*PLANIT FILE -- INC. LOGIN                                                  00037100
*PLANIT FILE -- INC. ADDEL                                                  00037200
*PLANIT FILE -- INC. ATTACH                                                 00037300
*PLANIT FILE -- INC. SAVE                                                   00037400
*PLANIT FILE -- INC. LOCKER                                                 00037500
*PLANIT FILE -- INC. RESTART                                                00037600
*PLANIT FILE -- INC. FRMERE                                                 00037700
*ENC                                                                        00037800
      END                                                                   00037900
*INS &ENDPROG                                                               00038000
      SUBROUTINE PLAN5                                                      00038100
*COMPCN                                                                     00038200
      IGOTO=IGOTC-<COMMAND-1>                                               00038300
      GO TO(<COMMAND>,1,<EDIT1>,<EDIT2>,<EDIT3>,<BUILD>,<REORDER>           00038400
     1   ,<SPOOL>),IGOTC                                                    00038500
2     IGOTO=0                                                               00038600
1     RETURN                                                                00038700
*PLANIT FILE -- INC. COMMAND                                                00038800
*PLANIT FILE -- INC. EDIT1                                                  00038900
*PLANIT FILE -- INC. EDIT2                                                  00039000
*PLANIT FILE -- INC. EDIT3                                                  00039100
*PLANIT FILE -- INC. BUILD                                                  00039200
*PLANIT FILE -- INC. REORDER                                                00039300
*PLANIT FILE -- INC. SPOOL                                                  00039400
*ENC                                                                        00039500
      END                                                                   00039600
*INS &ENDPROG                                                               00039700
      SUBROUTINE PLAN6                                                      00039800
*COMPCN                                                                     00039900
      IGOTO=IGOTC-<SYSTEM-1>                                                00040000
      GO TO(<SYSTEM>,1,<DELETE>,<UNLOAD>),IGOTC                             00040100
2     IGOTO=0                                                               00040200
1     RETURN                                                                00040300
*PLANIT FILE -- INC. SYSTEM                                                 00040400
*PLANIT FILE -- USE OCTAL CODE.                                             00040500
*PLANIT FILE -- USE OCTAL CODE.                                             00040600
*PLANIT FILE -- INC. DELETE                                                 00040700
*PLANIT FILE -- INC. UNLOAD                                                 00040800
*ENC                                                                        00040900
      END                                                                   00041000
*INS &ENDPROG                                                               00041100
      SUBROUTINE PLAN7                                                      00041200
*COMPCN                                                                     00041300
      IGOTO=IGOTC-<EDTEXT-1>                                                00041400
      GO TO(<EDTEXT>,1,<RENUM>,<PRESET>,1,<CFILE>                           00041500
     1   ,<INITIAL>),IGOTO                                                  00041600
2     IGOTO=0                                                               00041700
1     RETURN                                                                00041800
*PLANIT FILE -- INC. EDTEXT                                                 00041900
*PLANIT FILE -- INC. RENUM                                                  00042000
```

```
      *PLANIT FILE -- INC. PRESET                                                    CCC42100
      *PLANIT FILE -- INC. OFILE                                                     CCC42200
      *PLANIT FILE -- INC. INITIAL                                                   CCC42300
      *END                                                                          CCC42400
            END                                                                     CCC42500
      *INS GENCPROG                                                                 CCC42600
            SUBROUTINE PLANB                                                        CCC42700
      *COMMON                                                                       CCC42800
            IGOTO=IGOTO-<LOGOUT-1>                                                  CCC42900
            GO TO(<LOGOUT>,<SHOW>,<DISPLAY>,<ACCOUNT>                               CCC43000
           1  ,<WRITEP>,<CONFIRM>,<NODISK>,<FIX>),IGOTO                             CCC43100
      2     IGOTO=0                                                                 CCC43200
      1     RETURN                                                                  CCC43300
      *PLANIT FILE -- INC. LOGOUT                                                   CCC43400
      *PLANIT FILE -- INC. SHOW                                                     CCC43500
      *PLANIT FILE -- INC. DISPLAY                                                  CCC43600
      *PLANIT FILE -- INC. ACCOUNT                                                  CCC43700
      *PLANIT FILE -- INC. WRITEP                                                   CCC43800
      *PLANIT FILE -- INC. CONFIRM                                                  CCC43900
      *PLANIT FILE -- INC. NODISK                                                   CCC44000
      *PLANIT FILE -- INC. FIX                                                      CCC44100
      *END                                                                          CCC44200
            END                                                                     CCC44300
      *INS $$$$                                                                     CCC44400
      $$$$                                                                          CCC44500
```

APPENDIX D

Listing of the TRANSL program.

```
C:      PROGRAM TRANSL                                                  00000100
C:*************************************************************       00000200
C:THIS PROGRAM TRANSLATES THE FORTRAN PRODUCED BY THE PLANIT            00000300
C:GENERATOR INTO THE TACPOL LANGUAGE.  NO ADDITIONAL WORK               00000400
C:SHOULD BE NECESSARY ON THE RESULTING PLANIT CODE.  HOWEVER,           00000500
C:FOR TRANSLATING THE FORTRAN OF THE GENERATOR PROGRAM ITSELF,          00000600
C:OR THE FORTRAN IN THIS PROGRAM, INTO TACPOL, CERTAIN STATE-           00000700
C:MENTS MUST BE FIXED BY HAND, INCLUDING: READ, WRITE, FORMAT,          00000800
C:AND END FILE.  ALSO, A DIFFERENT OPTION MUST BE USED FOR              00000900
C:GENERATING THE DCL STATEMENTS SINCE NOT ALL OF THE ITEMS WILL         00001000
C:HAVE BEEN DECLARED IN COMMON AS THEY ARE IN THE PLANIT CODE.          00001100
C:SEE THE DESCRIPTION OF ITEM NTYPE, BELOW.                             00001200
C:                                                                      00001300
C:ITEM NTYPE IS A RUN OPTION PARAMETER WITH FOUR LEGAL VALUES           00001400
C:(1-4) SELECTING ONE OF THE FOLLOWING OPTIONS:                         00001500
C:                                                                      00001600
C:     1.  OUTPUT TACPOL CODE. DO NOT OUTPUT DCL STATEMENTS. FLAG       00001700
C:         ANY ITEMS FOUND IN THE CODE WHICH HAVE NOT BEEN DECLARED     00001800
C:         IN A PRIOR COMMON STATEMENT.                                 00001900
C:                                                                      00002000
C:     2.  OUTPUT TACPOL CODE. DO NOT OUTPUT DCL STATEMENTS. DO         00002100
C:         NOT FLAG UNDECLARED ITEMS WHICH ARE FOUND IN THE CODE.       00002200
C:                                                                      00002300
C:     3.  OUTPUT DCL STATEMENTS. DO NOT OUTPUT CODE. DO NOT FLAG       00002400
C:         UNDECLARED ITEMS WHICH ARE FOUND IN THE CODE.                00002500
C:                                                                      00002600
C:     4.  OUTPUT DCL STATEMENTS. DO NOT OUTPUT CODE. FLAG ANY          00002700
C:         ITEMS FOUND IN THE CODE WHICH HAVE NOT BEEN DECLARED         00002800
C:         IN A PRIOR COMMON STATEMENT.                                 00002900
C:                                                                      00003000
C:                                                                      00003100
C:THE TRANSLATOR WILL REPORT THREE KINDS OF ERRORS, ALL                 00003200
C:MIXED IN THE LINES OF THE OUTPUT FILE.  THEY WILL BE                  00003300
C:FLAGGED AND CAN USUALLY BE FIXED WITHOUT REPEATING                    00003400
C:THE JOB.  THE THREE TYPES OF ERRORS ARE:                              00003500
C:                                                                      00003600
C:     1.  A MESSAGE ON THE LAST LINE AND ABORT.  THIS                  00003700
C:         ERROR IS UNRECOVERABLE AND MUST BE FIXED AND                 00003800
C:         THE JOB RUN AGAIN.                                           00003900
C:                                                                      00004000
C:     2.  THE UNTRANSLATED INPUT LINE MAY BE PRINTED                   00004100
C:         IMMEDIATELY FOLLOWED BY A LINE OF SLASHES                    00004200
C:         (////////////////).  THIS INDICATES A REJECT                 00004300
C:         OF THE INPUT LINE FOR TRANSLATION.  AN EXAM-                 00004400
C:         INATION OF THE LINE WILL USUALLY REVEAL THE                  00004500
C:         PROBLEM, WHETHER A FORTRAN ERROR OR A STRING                 00004600
C:         THAT THE TRANSLATOR IS NOT EQUIPPED TO TRANS-                00004700
C:         LATE.  THIS KIND CAN OFTEN BE FIXED.                         00004800
C:                                                                      00004900
C:     3.  A MESSAGE MAY APPEAR WHICH SAYS:                             00005000
C:         UNDECLARED ITEM NAME ON THE NEXT LINE //////////             00005100
C:         WHICH CAN RESULT FROM CHOOSING NTYPE OPTION OF               00005200
C:         ONE OR FOUR.  THIS IS SPECIFICALLY FOR PLANIT                00005300
C:         WHERE ALL ITEMS ARE TO BE DECLARED IN COMMON AND             00005400
C:         THE TRANSLATOR HAS DETECTED A FAILURE.  THE                  00005500
C:         MESSAGE CAN BE DELETED AND IGNORED BUT THE EFFECT            00005600
C:         OF THE UNDECLARED ITEM TO PLANIT SHOULD BE CHECKED.          00005700
C:                                                                      00005800
C:AT THE END OF THE JOB, A STATEMENT WILL PRINT THE NUMBER              00005900
C:OF FLAGGED LINES (IF THERE ARE ANY).  THEY CAN THEN BE                00006000
```

```
C:SPOTTED EASILY IN THE LISTING OR BY USING A PROGRAM TO                  00006100
C:SEARCH FOR //// CHARACTERS.  IF THE PROBLEM INPUT LINE                 00006200
C:IS MORE THAN ONE CARD LONG, ONLY THE LAST CARD WILL BE                  00006300
C:PRINTED AND THE LISTING SHOULD BE CONSULTED FOR THE                     00006400
C:FULL LINE.                                                              00006500
C:                                                                        00006600
C:                                                                        00006700
      COMMON IFIRST(1)                                                    00006800
      COMMON IBLANK                                                       00006900
      COMMON MSGTBL(200)                                                  00007000
      COMMON JMRK                                                         00007100
      COMMON IBYT1                                                        00007200
      COMMON IBYT2                                                        00007300
      COMMON INDEX                                                        00007400
      COMMON LBYTE(500)                                                   00007500
      COMMON ICMPRE(500)                                                  00007600
      COMMON IFLINE(150)                                                  00007700
      COMMON INBUFF(30)                                                   00007800
      COMMON INQUE(30)                                                    00007900
      COMMON ICHK(6)                                                      00008000
      COMMON LABOO(5)                                                     00008100
      COMMON LISTN(150)                                                   00008200
      COMMON KARDUM                                                       00008300
      COMMON KARSET(58)                                                   00008400
      COMMON MAPDUM                                                       00008500
      COMMON MAP(255)                                                     00008600
      COMMON MSGDR1(90)                                                   00008700
      COMMON MSGDR2(90)                                                   00008800
      COMMON JFLG                                                         00008900
      COMMON NPAREN(5)                                                    00009000
      COMMON KINDST(5)                                                    00009100
      COMMON NFN1(5)                                                      00009200
      COMMON NFN2(5)                                                      00009300
C:*************************************************************          00009400
C:THE FOLLOWING STATEMENTS (BETWEEN PAIRS OF ASTERISKS)                   00009500
C:COULD VARY ON A DIFFERENT COMPUTER.  ALSO, THE READ, WRITE,             00009600
C:FORMAT AND END FILE STATEMENTS WILL HAVE TO BE CHANGED FOR              00009700
C:TRANSLATION INTO TACPOL.                                                00009800
C:                                                                        00009900
C:BYTES PER WORD OF THE COMPUTER RUNNING THIS PROGRAM.                    00010000
      IBYTE=4                                                             00010100
      GO TO 9                                                             00010200
C:                                                                        00010300
1000  FORMAT(20A4)                                                        00010400
C:                                                                        00010500
1     CONTINUE                                                            00010600
C:END OF THE TRANSLATE JOB. END FILE STATEMENT ON NEXT                    00010700
C:LINE CLOSES THE OUTPUT FILE. CHANGE OR DELETE IT IF                     00010800
C:NECESSARY                                                               00010900
      END FILE 2                                                          00011000
      GO TO 198                                                           00011100
2     CONTINUE                                                            00011200
      READ(1,1000) (INBUFF(I), I=1,NUMBUF)                               00011300
      GO TO 4                                                             00011400
3     CONTINUE                                                            00011500
      WRITE(2,1000) (INBUFF(I), I=1,NUMBUF)                               00011600
      GO TO 183                                                           00011700
C:*************************************************************          00011800
4     CONTINUE                                                            00011900
C:QUEUE INTO BUFFER.                                                      00012000
```

```
      DO 5 I=1,NUMBUF                                                   00012100
      J=INBUFF(I)                                                       00012200
      INBUFF(I)=INQUE(I)                                                00012300
      INQUE(I)=J                                                        00012400
5     CONTINUE                                                          00012500
      IERR=0                                                            00012600
      IBYT1=IBUFF+INCRE                                                 00012700
      IBYT2=IBUFF+71                                                    00012800
      INDEX=IWLBYT+NREAD                                                00012900
      CALL LDBYTE                                                       00013000
      IF(JSW .LE. 0) GO TO 11                                           00013100
      NREAC=NREAD-INCRE+72                                              00013200
6     CONTINUE                                                          00013300
C:MARK THE END OF THE INPUT LINE.                                       00013400
      IF(LBYTE(NREAD) .NE. KARSET(50)) GO TO 8                          00013500
      NREAD=NREAD-1                                                     00013600
      IF(NREAD .GT. 0) GO TO 6                                          00013700
7     CONTINUE                                                          00013800
      NREAC=0                                                           00013900
      IF(ICHK(1) .EQ. KARSET(55)) GO TO 189                             00014000
      LSTCHK=KARSET(47)                                                 00014100
      GO TO 2                                                           00014200
8     CONTINUE                                                          00014300
      IF(JSW .EQ. 1) GO TO 13                                           00014400
C:BUFFERS LOADED AND END FOUND. CHECK FOR CONTINUATIONS                 00014500
C:AND IF LABELS.                                                        00014600
      IBYT1=IBQUE                                                       00014700
      IBYT2=IBYT1+5                                                     00014800
      INDEX=IWCHK                                                       00014900
      CALL LDBYTE                                                       00015000
      I=ICHK(1)                                                         00015100
      IF(MAP(I) .NE. 5C) GO TO 38                                       00015200
C:MAY BE CONTINUATION CARD ONLY IF BLANK IN COLUMN ONE.                 00015300
      J=ICHK(6)                                                         00015400
      IF(MAP(J) .GT. 35) GO TO 38                                       00015500
C:CONTINUATION CARD IF NOT TRUE.                                        00015600
      IF(NREAD+66 .GE. MAXLIN) GO TO 152                                00015700
      INCRE=6                                                           00015800
      GO TO 2                                                           00015900
9     CONTINUE                                                          00016000
C:INITIALIZATION SECTION.                                               00016100
C:IN THE RARE EVENT THAT ARRAY DIMENSIONS ARE TO BE                     00016200
C:CHANGED, THE FOLLOWING ASSIGNMENTS MUST ALSO BE                       00016300
C:CHANGED CORRESPONDINGLY.                                              00016400
C:                                                                      00016500
C:DIMENSION OF MSGTBL:                                                  00016600
      MSGDIM=200                                                        00016700
C:SIZE OF WORKING LINE (DIMENSIONS OF LBYTE AND ICMPRE):                00016800
      MAXLIN=500                                                        00016900
C:DIMENSION OF IFLINE:                                                  00017000
      IFLDIM=150                                                        00017100
C:BUFFER LENGTHS (DIMENSION OF INBUFF AND INQUE):                       00017200
      IBFDIM=30                                                         00017300
C:SIZE OF ITEM DICTIONARY (ONE-THIRD OF DIMENSION OF LISTN):            00017400
      MAXNAM=500                                                        00017500
C:END OF ARRAY DIMENSION ASSIGNMENTS.  ARRAYS BELOW LISTN               00017600
C:MAY HAVE DIMENSIONS CHANGED, IF NECESSARY, WITHOUT THE                00017700
C:NEED FOR ADJUSTMENTS HERE.                                            00017800
C:                                                                      00017900
C:WORD NUMBER ASSIGNMENTS.                                              00018000
```

```
      IWLBYT=MSGDIM+7                                                    00018100
      IWCMP=IWLBYT+MAXLIN                                                00018200
      IWBUFF=IWCMP+MAXLIN+IFLDIM                                         00018300
      IWQUE=IWBUFF+IBFDIM                                                00018400
      IWCHK=IWQUE+IBFDIM                                                 00018500
C:BYTE NUMBER ASSIGNMENTS.                                               00018600
      IBUFF=IWBUFF*IBYTE-IBYTE+1                                         00018700
      IWCMP=IWCMP*IBYTE-IBYTE+1                                          00018800
      IBQUE=IWQUE*IBYTE-IBYTE+1                                          00018900
C:                                                                       00019000
      IFIRST(1)=IWLBYT                                                   00019100
      DO 10 I=1,255                                                      00019200
      MAP(I)=58                                                          00019300
10    CONTINUE                                                           00019400
      MAPDUM=58                                                          00019500
      LINE=0                                                             00019600
      INCRE=0                                                            00019700
      INTRY=0                                                            00019800
      MSGN=0                                                             00019900
      JMPK=0                                                             00020000
      JFLG=0                                                             00020100
      NUMERR=0                                                           00020200
      NUMCO=0                                                            00020300
      IFRTRN=0                                                           00020400
      JSW=-1                                                             00020500
      IREAD=1                                                            00020600
      NTYPE=1                                                            00020700
      NUMBUF=79/IBYTE+1                                                  00020800
      GO TO 7                                                            00020900
11    CONTINUE                                                           00021000
C:SET UP TRANSLATION TABLES.                                             00021100
      JSW=JSW+1                                                          00021200
      IF(JSW .LE. 0) GO TO 2                                             00021300
      DO 12 I=1,56                                                       00021400
      J=LBYTE(I)                                                         00021500
      KARSET(I-1)=J                                                      00021600
      MAP(J)=I-1                                                         00021700
      ICMPRE(I)=LBYTE(51)                                                00021800
12    CONTINUE                                                           00021900
C:STORE BLANKS.                                                          00022000
      IBYT1=IBYTE+1                                                      00022100
      IBYT2=2*IBYTE                                                      00022200
      INDEX=IWCMP                                                        00022300
      CALL SBYTE                                                         00022400
      GO TO 7                                                            00022500
13    CONTINUE                                                           00022600
      I=LBYTE(1)                                                         00022700
      I=MAP(I)                                                           00022800
      IF(I .LE. 9) GO TO 14                                              00022900
      IF(I .EQ. 55) GO TO 15                                             00023000
      IF(I .NE. 50) GO TO 7                                              00023100
C:STORE INITIALIZATION INPUT DECK.                                       00023200
C:STORE MESSAGES.                                                        00023300
      IBYT1=MSGDR2(MSGN)+1                                               00023400
      IF(MSGN .EQ. 0) IBYT1=2*IBYTE+1                                    00023500
      IBYT2=IBYT1+NREAD-3                                                00023600
      INDEX=IWLBYT+2                                                     00023700
      CALL SBYTE                                                         00023800
      MSGN=MSGN+1                                                        00023900
      MSGDR1(MSGN)=IBYT1                                                 00024000
```

```
      MSGDR2(MSGN)=IBYT2                                                  00024100
      IF(JMRK .EQ. JFLG) GO TO 7                                          00024200
      ICHK(1)=KARSET(55)                                                  00024300
      N5=40                                                               00024400
      GO TO 196                                                           00024500
14    CONTINUE                                                            00024600
C:STORE TRANSLATION OPTION PARAMETER.                                     00024700
      NTYPE=1                                                             00024800
      GO TO 7                                                             00024900
15    CONTINUE                                                            00025000
      JSW=2                                                               00025100
      IF(NTYPE .NE. 3) GO TO 7                                            00025200
      N5=79                                                               00025300
      GO TO 196                                                           00025400
16    CONTINUE                                                            00025500
C:MARK OFF ITEM NAME AND COMPARE STRINGS TO PRESTORED                     00025600
C:NAMES. START AT LOOP TO N1. N2, N3 MARK OFF LIST OF                     00025700
C:WORDS TO BE COMPARED. N2 MARKS THE FOUND NAME.                          00025800
C:N4 IS THE RETURN CODE.                                                  00025900
      N1=LOOP                                                             00026000
17    CONTINUE                                                            00026100
      N1=N1+1                                                             00026200
      IF(N1 .GT. NREAD) GO TO 18                                          00026300
      I=LBYTE(N1)                                                         00026400
      IF(MAP(I) .LE. 35) GO TO 17                                         00026500
18    CONTINUE                                                            00026600
      N1=N1-1                                                             00026700
C:CHECK BASIC WORD LIST.                                                  00026800
19    CONTINUE                                                            00026900
      IBYT1=MSGDR1(N2)                                                    00027000
      IBYT2=MSGDR2(N2)                                                    00027100
      IF(IBYT2-IBYT1 .NE. N1-LOOP) GO TO 27                               00027200
C:CHARACTER COUNT OK.                                                     00027300
      INDEX=INCMP+LOOP-1                                                  00027400
      CALL LDBYTE                                                         00027500
      DO 20 I=LOOP,N1                                                     00027600
      IF(LBYTE(I) .NE. ICMPRE(I)) GO TO 27                                00027700
20    CONTINUE                                                            00027800
C:MATCH FOUND.                                                            00027900
      IF(N2 .LE. 25) GO TO 54                                             00028000
C:MATCH ON COMMAND WORD.                                                  00028100
      IF(N3 .EQ. 38) GO TO 125                                            00028200
C:MATCH ON RELATIONAL.                                                    00028300
      IF(N3 .EQ. 70) GO TO 55                                             00028400
C:MATCH ON CALL NAME.                                                     00028500
C:OTHERWISE CHECK FOR NAME CHANGE.                                        00028600
      IF(N2 .EQ. 26) LBYTE(LOOP)=KARSET(33)                               00028700
C:VALUE TO XVALUE.                                                        00028800
      IF(N2 .NE. 27) GO TO 21                                             00028900
C:L TO H.                                                                 00029000
      LBYTE(LOOP)=KARSET(17)                                              00029100
      GO TO 22                                                            00029200
21    CONTINUE                                                            00029300
C:CHECK WORD AGAINST NEW-WORD LIST.                                       00029400
      I=LBYTE(LOOP)                                                       00029500
      IF(MAP(I) .GE. 18 .AND. MAP(I) .LE. 23) GO TO 22                    00029600
C:FLOATING ITEM.                                                          00029700
      IF(IEQ .EQ. 0) KINDL=1                                              00029800
      IF(IEQ .NE. 0) KINDR=1                                              00029900
```

```
22      CONTINUE                                                        00030100
        IF(N4 .NE. 1 .OR. IEQ .NE. C) GO TO 26                          00030200
        ICMPRE(1)=IBLANK                                                00030300
        ICMPRE(2)=IBLANK                                                00030400
        ICMPRE(3)=IBLANK                                                00030500
        IBYT1=IBCMP                                                     00030600
        IBYT2=IBYT1+N1-LOOP                                             00030700
        INDEX=IWLBYT+LOOP-1                                             00030800
        CALL SBYTE                                                      00030900
        IF(INTRY .EQ. 0) GO TO 24                                       00031000
        J=3*INTRY                                                       00031100
        DO 23 I=1,J,3                                                   00031200
        IF(ICMPRE(1) .EQ. LISTN(I) .AND.                                00031300
     1     ICMPRE(2) .EQ. LISTN(I+1) .AND.                              00031400
     2     ICMPRE(3) .EQ. LISTN(I+2)) GO TO 26                          00031500
23      CONTINUE                                                        00031600
24      CONTINUE                                                        00031700
        INTRY=INTRY+1                                                   00031800
        IF(INTRY .LE. MAXNAM) GO TO 25                                  00031900
C:TOO MANY ITEM NAME ENTRIES. PRINT ERROR MSG AND STOP.                 00032000
        ICHK(1)=KARSET(55)                                              00032100
        N5=64                                                           00032200
        GO TO 196                                                       00032300
25      CONTINUE                                                        00032400
C:ITEM NAME AS YET UNDECLARED.                                          00032500
        J=INTRY*3                                                       00032600
        LISTN(J-2)=ICMPRE(1)                                            00032700
        LISTN(J-1)=ICMPRE(2)                                            00032800
        LISTN(J)=ICMPRE(3)                                              00032900
        IF(KCMMON .NE. 0) GO TO 142                                     00033000
C:UNDECLARED NAME FOUND IN THE CODE.                                    00033100
        IF(NTYPE .EQ. 1 .OR. NTYPE .EQ. 4) GO TO 139                    00033200
        IF(N4 .EQ. 1) GO TO 142                                         00033300
26      CONTINUE                                                        00033400
        GO TO(45,62,66,111),N4                                          00033500
27      CONTINUE                                                        00033600
        N2=N2+1                                                         00033700
        IF(N2 .LE. N3) GO TO 15                                         00033800
C:NO MATCH.                                                             00033900
        IF(N3 .EQ. 38) GO TO 126                                        00034000
C:IF NO MATCH ON A RELATIONAL, TRY IT AS A NUMBER.                      00034100
        IF(N3 .EQ. 70) GO TO 164                                        00034200
C:IF NOT ONE OF THE LISTED CALL NAMES, THEN WRITE                       00034300
C:THE LINE AS IS.                                                       00034400
        GO TO 21                                                        00034500
28      CONTINUE                                                        00034600
C:MARK OFF NUMBER BEGINNING AT LOOP, MARK END AT N1.                    00034700
C:NUMBER VALUE IN N2 AND N3 IS 0 FOR INTEGER, NOT 0 FOR                 00034800
C:FLOATING. N4 IS THE RETURN CODE.                                      00034900
        N3=0                                                            00035000
        N1=LOOP                                                         00035100
        N2=LBYTE(N1)                                                    00035200
        N2=MAP(N2)                                                      00035300
        IF(N2 .LE. 9) GO TO 29                                          00035400
        N2=0                                                            00035500
        N3=LOOP                                                         00035600
29      CONTINUE                                                        00035700
        N1=N1+1                                                         00035800
        IF(N1 .GT. NREAD) GO TO 31                                      00035900
        I=LBYTE(N1)                                                     00036000
```

```
          I=MAP(I)                                                      00036100
          IF(I .NE. 42) GO TO 30                                        00036200
C:DECIMAL POINT.                                                        00036300
          IF(N3 .NE. 0) GO TO 32                                        00036400
          N3=N1                                                         00036500
          GO TO 29                                                      00036600
30        CONTINUE                                                      00036700
          IF(I .GT. 9) GO TO 31                                         00036800
          N2= 10*N2+I                                                   00036900
          GO TO 29                                                      00037000
31        CONTINUE                                                      00037100
C:END OF NUMBER.                                                        00037200
          N1=N1-1                                                       00037300
          IF(N1 .NE. N3) GO TO 33                                       00037400
C:NUMBER ENDS IN DECIMAL POINT.                                         00037500
          IF(N1 .EQ. NREAD) GO TO 33                                    00037600
          I=LBYTE(N1+1)                                                 00037700
          IF(MAP(I) .LT. 10 .OR. MAP(I) .GT. 35)                        00037800
     1       GO TO 33                                                   00037900
C:DECIMAL GOES WITH SOMETHING ELSE.                                     00038000
          N3=0                                                          00038100
32        CONTINUE                                                      00038200
          N1=N1-1                                                       00038300
33        CONTINUE                                                      00038400
          IF(N3 .EQ. 0) GO TO 37                                        00038500
C:NUMBER CONTAINS DECIMAL POINT.                                        00038600
          IF(LOOP .EQ. N1) GO TO 192                                    00038700
C:DECIMAL POINT ONLY IF TRUE.                                           00038800
          IF(IEQ .EQ. 0) KINDL=1                                        00038900
          IF(IEQ .NE. 0) KINDR=1                                        00039000
34        CONTINUE                                                      00039100
          N1=N1+1                                                       00039200
          NREAD=NREAD+1                                                 00039300
          I=NREAD                                                       00039400
35        CONTINUE                                                      00039500
          I=I-1                                                         00039600
          IF(I .LT. N1) GO TO 36                                        00039700
          LBYTE(I+1)=LBYTE(I)                                           00039800
          GO TO 35                                                      00039900
C:INSERT L AT END OF LITERAL FLOATING POINT NUMBER.                     00040000
36        CONTINUE                                                      00040100
          LBYTE(N1)=KARSET(21)                                          00040200
C:INSERT TRAILING ZERO IF NECESSARY.                                    00040300
          IF(N1 .NE. N3+1) GO TO 37                                     00040400
          LBYTE(N1)=KARDUM                                              00040500
          GO TO 34                                                      00040600
37        CONTINUE                                                      00040700
          GO TO(40,55,66,45),N4                                         00040800
38        CONTINUE                                                      00040900
C:FORTRAN LINE READY. SCAN FROM COLUMN ONE.                             00041000
          INCRE=0                                                       00041100
          IBGN=0                                                        00041200
          KINDL=0                                                       00041300
          KINDR=0                                                       00041400
          IDPTH=0                                                       00041500
          IEQ=0                                                         00041600
          IFST=0                                                        00041700
          KOMMON=0                                                      00041800
          JMRK=0                                                        00041900
          NUMEND=0                                                      00042000
```

```
      ICHR=LBYTE(1)                                                     00042100
      IF(MAP(ICHR) .NE. 12) GO TO 39                                    00042200
C:COMMENT. BRACKET WITH CHARACTERS.                                     00042300
      LBYTE(1)=KARSET(39)                                               00042400
      LBYTE(2)=KARSET(38)                                               00042500
      LBYTE(70)=KARSET(36)                                              00042600
      LBYTE(71)=KARSET(35)                                              00042700
      NREAD=71                                                          00042800
      GO TO 165                                                         00042900
39    CONTINUE                                                          00043000
      IF(MAP(ICHR) .GT. 9) GO TO 42                                     00043100
C:LABEL. PREFIX A 0 AND SUFFIX A COLON.                                 00043200
      LOOP=1                                                            00043300
      N4=1                                                              00043400
      GO TO 28                                                          00043500
40    CONTINUE                                                          00043600
      IF(LBYTE(N1+1) .NE. KARSET(50)) GO TO 192                         00043700
      LBYTE(N1+1)=KARSET(46)                                            00043800
C:CHECK DO FOR PROPER END STATEMENTS.                                   00043900
41    CONTINUE                                                          00044000
      IF(NUMDO .EQ. 0 .OR. LABDO(NUMDO) .NE. N2) GO TO 44               00044100
C:COUNT NUMBER OF ENDS TO APPEND.                                       00044200
      NUMDO=NUMDO-1                                                     00044300
      NUMEND=NUMEND+1                                                   00044400
      GO TO 41                                                          00044500
42    CONTINUE                                                          00044600
      IF(MAP(ICHR) .EQ. 50) GO TO 43                                    00044700
C:DON'T TRANSLATE LINE. DELETE '8' PREFIX.                              00044800
      IF(NTYPE .GE. 3) GO TO 7                                          00044900
      IF(MAP(ICHR) .EQ. 11) LBYTE(1)=KARSET(50)                         00045000
      GO TO 193                                                         00045100
43    CONTINUE                                                          00045200
C:FORTRAN STATEMENT FIELD.                                              00045300
      IF(LBYTE(6) .NE. KARSET(50)) GO TO 192                            00045400
44    CONTINUE                                                          00045500
      N1=6                                                              00045600
45    CONTINUE                                                          00045700
      N5=1                                                              00045800
46    CONTINUE                                                          00045900
      LOOP=N1+1                                                         00046000
C:SCAN VARIABLE FOR CRACKING THE FORTRAN LINE.                          00046100
      IF(LOOP .LE. NREAD) GO TO 51                                      00046200
C:END OF FORTRAN LINE REACHED.                                          00046300
      IF(N5 .NE. 1) GO TO 192                                           00046400
      IF(KOMMON .NE. 0) GO TO 7                                         00046500
      IF(IEQ .EQ. 0 .OR. KINDL .EQ. KINDR) GO TO 49                     00046600
C:FLOAT TO FIX CONVERSION (OR VICE VERSA) NECESSARY.                    00046700
      N1=IEQ+1                                                          00046800
      N2=0                                                              00046900
      N3=2                                                              00047000
      N4=KINDL+51                                                       00047100
      N5=10                                                             00047200
      GO TO 152                                                         00047300
47    CONTINUE                                                          00047400
      N1=NREAD+1                                                        00047500
      N2=0                                                              00047600
      N4=KINDL+53                                                       00047700
      IF(NUMEND .EQ. 0) GO TO 73                                        00047800
48    CONTINUE                                                          00047900
      N3=2                                                              00048000
```

```
      N5=1                                                              00048100
      GO TO 152                                                         00048200
C:APPEND ENDS.                                                          00048300
49    CONTINUE                                                          00048400
      IF(NUMEND .EQ. 0) GO TO 164                                       00048500
      N4=45                                                             00048600
50    CONTINUE                                                          00048700
      NUMEND=NUMEND-1                                                   00048800
      N1=NREAD+1                                                        00048900
      N2=0                                                              00049000
      GO TO 48                                                          00049100
51    CONTINUE                                                          00049200
      N1=LOOP                                                           00049300
      I=LBYTE(LOOP)                                                     00049400
      I=MAP(I)                                                          00049500
      IF(I .EQ. 50) GO TO 46                                            00049600
C:SKIP BLANKS.                                                          00049700
      GO TO(52,61,63,65,66,56,75                                        00049800
     1     ,83,106,80,143,145,98),N5                                    00049900
C:LOOP VARIABLE OF SCANNER SET TO NEXT NON-BLANK.                       00050000
52    CONTINUE                                                          00050100
      IF(IBGN .NE. 0) GO TO 125                                         00050200
      IF(I .LT. 10 .OR. I .GT. 35) GO TO 192                            00050300
      IBGN=LOOP                                                         00050400
C:FIRST CHARACTER IN STATEMENT FIELD MUST BE A LETTER.                  00050500
      N2=1                                                              00050600
53    CONTINUE                                                          00050700
      N3=30                                                             00050800
      N4=1                                                              00050900
      GO TO 16                                                          00051000
54    CONTINUE                                                          00051100
C:BASIC WORD LIST SWITCH.  SWITCH POINTS ARE FOR THE FOLLOWING:         00051200
C:   CONTINUE,DO,IF,COMMON,CALL,SUBROUTINE,FORMAT,READ,WRITE,PAUSE,    00051300
C:   STOP,PROC,END,ENDFILE,GO,GOTO,ABS,IABS,ALOG,SQRT,SIN,CCS,         00051400
C:   RETURN.                                                            00051500
      GO TO(55,57,74,96,97,101,152                                      00051600
     1     ,152,152,150,55,164,145,152                                  00051700
     2     ,103,105,117,116,115,115                                     00051800
     3     ,120,120,150),N2                                             00051900
55    CONTINUE                                                          00052000
C:CONTINUE OR STOP. ERASE IT.                                           00052100
      IF(N1 .NE. NREAD) GO TO 152                                       00052200
      DO 56 I=LOOP,N1                                                   00052300
      LBYTE(I)=KARSET(50)                                               00052400
56    CONTINUE                                                          00052500
      IF(NUMEND .EQ. 0 .OR. N2 .NE. 1) GO TO 165                        00052600
      NREAD=LOOP-1                                                      00052700
      N4=13                                                             00052800
      GO TO 50                                                          00052900
57    CONTINUE                                                          00053000
C: DO STATEMENT.                                                        00053100
      N5=6                                                              00053200
      GO TO 46                                                          00053300
58    CONTINUE                                                          00053400
      IF(I .GT. 5) GO TO 152                                            00053500
      N4=2                                                              00053600
      GO TO 28                                                          00053700
C:STORE THE DO LABEL.                                                   00053800
59    CONTINUE                                                          00053900
      DO 60 I=LOOP,N1                                                   00054000
```

```
      LBYTE(I)=KARSET(50)                                                00054100
60    CONTINUE                                                           00054200
      NUMDO=NUMDO+1                                                      00054300
      LABDO(NUMDO)=N2                                                    00054400
      N5=2                                                               00054500
      GO TO 46                                                           00054600
C:CHECK DO NAME FOR POSSIBLE CONFLICT.                                   00054700
61    CONTINUE                                                           00054800
      IF(I .LT. 10 .OR. I .GT. 35) GO TO 192                             00054900
      N2=26                                                              00055000
      N3=30                                                              00055100
      N4=2                                                               00055200
      GO TO 16                                                           00055300
62    CONTINUE                                                           00055400
      N5=3                                                               00055500
C:CHECK FOR EQUAL SIGN.                                                  00055600
      GO TO 46                                                           00055700
63    CONTINUE                                                           00055800
      IF(I .NE. 44) GO TO 152                                            00055900
64    CONTINUE                                                           00056000
      N5=4                                                               00056100
      GO TO 46                                                           00056200
65    CONTINUE                                                           00056300
      IF(I .GT. 35) GO TO 152                                            00056400
      N4=3                                                               00056500
      IF(I .LT. 10) GO TO 28                                             00056600
      N2=26                                                              00056700
      N3=30                                                              00056800
      GO TO 16                                                           00056900
66    CONTINUE                                                           00057000
      IF(JMRK) 71,67,70                                                  00057100
67    CONTINUE                                                           00057200
      N5=5                                                               00057300
C:CHECK COMMA IN LABEL LIST.                                             00057400
      GO TO 46                                                           00057500
68    CONTINUE                                                           00057600
      IF(I .NE. 45) GO TO 152                                            00057700
C:INSERT BY 1 TO                                                         00057800
      LBYTE(LOOP)=KARSET(50)                                             00057900
      IF(JMRK .NE. 0) GO TO 64                                           00058000
      JMRK=LOOP+4                                                        00058100
      N2=0                                                               00058200
      N3=2                                                               00058300
      N4=44                                                              00058400
      N5=5                                                               00058500
      GO TO 152                                                          00058600
69    CONTINUE                                                           00058700
      N1=LOOP+8                                                          00058800
      GO TO 64                                                           00058900
70    CONTINUE                                                           00059000
      IF(N1 .EQ. NREAD) GO TO 164                                        00059100
      JMRK=-JMRK                                                         00059200
      GO TO 67                                                           00059300
71    CONTINUE                                                           00059400
      IF(N1 .NE. NREAD) GO TO 192                                        00059500
      IBYT1=IBCMP+30                                                     00059600
      IBYT2=IBYT1+N1-LOOP                                                00059700
      INDEX=IWLBYT+LOOP-1                                                00059800
      CALL SBYTE                                                         00059900
      DO 72 I=LOOP,N1                                                    00060000
```

```
      LBYTE(I1)=KARSET(50)                                              00060100
72    CONTINUE                                                          00060200
      NREAD=LOOP-2                                                      00060300
C: PUT THIRD DO PARAMETER IN PLACE OF CURRENT BY VALUE.                 00060400
      MSGDR1(40)=IBYT1                                                  00060500
      MSGDR2(40)=IBYT2                                                  00060600
      N1=IABS(JMRK)                                                     00060700
      N2=41                                                             00060800
      N4=40                                                             00060900
73    CONTINUE                                                          00061000
      N3=2                                                              00061100
      GO TO 95                                                          00061200
74    CONTINUE                                                          00061300
C: IF STATEMENT.                                                        00061400
      IDPTH=IDPTH+1                                                     00061500
      IF(IDPTH .GT. 5) GO TO 192                                        00061600
      KINDST(IDPTH)=1                                                   00061700
75    CONTINUE                                                          00061800
      NPAREN(IDPTH)=-1                                                  00061900
      GO TO 45                                                          00062000
76    CONTINUE                                                          00062100
C: LEFT PAREN.                                                          00062200
      IF(IDPTH .EQ. 0) GO TO 46                                         00062300
      IF(NPAREN(IDPTH)) 77,192,78                                       00062400
77    CONTINUE                                                          00062500
      NPAREN(IDPTH)=0                                                   00062600
78    CONTINUE                                                          00062700
      NPAREN(IDPTH)=NPAREN(IDPTH)+1                                     00062800
      GO TO 46                                                          00062900
79    CONTINUE                                                          00063000
C: RIGHT PAREN.                                                         00063100
      LSTPRN=LOOP                                                       00063200
      IF(IDPTH .EQ. 0) GO TO 46                                         00063300
      IF(NPAREN(IDPTH) .LE. 0) GO TO 192                                00063400
      NPAREN(IDPTH)=NPAREN(IDPTH)-1                                     00063500
      IF(NPAREN(IDPTH) .NE. 0) GO TO 46                                 00063600
C: FOUND THE MATCHING CLOSE PAREN.                                      00063700
      IF(KINDST(IDPTH) .GE. 3) GO TO 121                                00063800
      N5=10                                                             00063900
      IF(N1 .LT. NREAD) GO TO 46                                        00064000
      GO TO 192                                                         00064100
80    CONTINUE                                                          00064200
C: END OF IF CLAUSE.                                                    00064300
      IF(I .GT. 9) KINDST(IDPTH)=2                                      00064400
C: INSERT THEN CLAUSE.                                                  00064500
      N2=0                                                              00064600
      N3=2                                                              00064700
      N4=46                                                             00064800
      N5=3                                                              00064900
      GO TO 152                                                         00065000
81    CONTINUE                                                          00065100
      N1=LOOP+4                                                         00065200
      LBYTE(N1)=KARSET(50)                                              00065300
      IFST=N1                                                           00065400
      IEQ=0                                                             00065500
      JMRK=KINDL                                                        00065600
      KINDL=0                                                           00065700
      KINDR=0                                                           00065800
      IF(IDPTH .NE. 1) GO TO 192                                        00065900
      IDPTH=0                                                           00066000
```

```
C:RESET POINTERS FOR STATEMENT FOLLOWING IF STATEMENT.                  00066100
      IF(KINDST(1) .NE. 1) GO TO 45                                     00066200
C:IF ABOVE NOT TRUE, THEN IT IS AN ARITHMETIC IF FORM.                  00066300
      J=0                                                               00066400
      DO 82 I=1,5                                                       00066500
      K=ICHK(I)                                                         00066600
      IF(MAP(K) .EQ. 50) GO TO 82                                       00066700
      IF(MAP(K) .GT. 9) GO TO 152                                       00066800
      J=10*J+MAP(K)                                                     00066900
82    CONTINUE                                                          00067000
      IF(J .EQ. 0) GO TO 152                                            00067100
C:J IS THE NEXT LABEL AFTER THE IF.                                     00067200
      ICHK(3)=J                                                         00067300
      ICHK(4)=0                                                         00067400
      ICHK(5)=0                                                         00067500
      ICHK(6)=0                                                         00067600
      JCNT=4                                                            00067700
      N5=8                                                              00067800
      GO TO 46                                                          00067900
83    CONTINUE                                                          00068000
C:COLLECT LABEL NUMBERS.                                                00068100
      IF(I .NE. 45) GO TO 84                                            00068200
      JCNT=JCNT+1                                                       00068300
      I=0                                                               00068400
84    CONTINUE                                                          00068500
      IF(I .GT. 9) GO TO 152                                            00068600
      ICHK(JCNT)=10*ICHK(JCNT)+I                                        00068700
      IF(N1 .LT. NREAD) GO TO 46                                        00068800
      IF(JCNT .NE. 6) GO TO 152                                         00068900
C:LABELS NOW ARE STORED IN ICHK.                                        00069000
      DO 85 I=IFST,NREAD                                                00069100
      LBYTE(I)=KARSET(50)                                               00069200
85    CONTINUE                                                          00069300
      NREAD=IFST                                                        00069400
C:REMOVE LABELS FROM STATEMENT.                                         00069500
      N1=LSTPRN                                                         00069600
      N4=JMRK+47                                                        00069700
C:INSERT COMPARISON WITH ZERO INTO STATEMENT.                           00069800
86    CONTINUE                                                          00069900
      N2=0                                                              00070000
      N3=2                                                              00070100
      N5=4                                                              00070200
      JCNT=N1                                                           00070300
      GO TO 152                                                         00070400
87    CONTINUE                                                          00070500
      IF(JCNT .NE. LSTPRN) GO TO 88                                     00070600
      JMRK=NREAD                                                        00070700
C:INSERT GOTO                                                           00070800
      N1=NREAD+1                                                        00070900
      N4=16                                                             00071000
      GO TO 86                                                          00071100
88    CONTINUE                                                          00071200
      LSTPRN=LSTPRN+1                                                   00071300
      IFST=NREAD+1                                                      00071400
      IF(IFST .GT. 150) GO TO 152                                       00071500
      DO 89 I=1,150                                                     00071600
      IFLINE(I)=LBYTE(I)                                                00071700
89    CONTINUE                                                          00071800
      JFLG=0                                                            00071900
      JCNT=3                                                            00072000
```

```
90      CONTINUE                                                                00072100
        JSW=2                                                                   00072200
        JCNT=JCNT+1                                                             00072300
        IF(JCNT .GT. 6) GO TO 7                                                 00072400
C:CHECK IF LABEL WITH NEXT LINE LABEL.                                          00072500
        IF(ICHK(3) .NE. ICHK(JCNT)) GO TO 91                                    00072600
        JFLG=1                                                                  00072700
        GO TO 90                                                                00072800
91      CONTINUE                                                                00072900
C:RESTORE LINE INTO LBYTE.                                                      00073000
        DO 92 I=1,150                                                           00073100
        LBYTE(I)=IFLINE(I)                                                      00073200
        IF(I .LE. 5) IFLINE(I)=KARSET(50)                                       00073300
92      CONTINUE                                                                00073400
        NREAD=IFST                                                              00073500
C:TEST FOR SIMPLE GOTO DEFAULT CASE.                                            00073600
        IF(JCNT .EQ. 6 .AND. JFLG .EQ. 0) GO TO 93                              00073700
C:INSERT PROPER RELATIONAL.                                                     00073800
        N1=LSTPRN                                                               00073900
        N2=N1+1                                                                 00074000
        N3=2                                                                    00074100
        N4=JCNT+37                                                              00074200
        N5=6                                                                    00074300
        GO TO 152                                                               00074400
93      CONTINUE                                                                00074500
        N1=7                                                                    00074600
        N2=JMRK                                                                 00074700
        N3=0                                                                    00074800
        N5=6                                                                    00074900
        GO TO 152                                                               00075000
94      CONTINUE                                                                00075100
C:APPEND FINAL LABEL.                                                           00075200
        N1=NREAD+1                                                              00075300
        N2=0                                                                    00075400
        N3=1                                                                    00075500
        N4=-ICHK(JCNT)                                                          00075600
        JSW=3                                                                   00075700
95      CONTINUE                                                                00075800
        N5=7                                                                    00075900
        GO TO 152                                                               00076000
96      CONTINUE                                                                00076100
C:COMMON STATEMENT.                                                             00076200
        KOMMON=1                                                                00076300
        GO TO 45                                                                00076400
97      CONTINUE                                                                00076500
        IF(NTYPE .NE. 1) GO TO 164                                              00076600
C:CALL STATEMENT. CHECK FOR LDBYTE, SBYTE AND MIOP CALLS.                       00076700
        N5=13                                                                   00076800
        GO TO 46                                                                00076900
98      CONTINUE                                                                00077000
        N2=66                                                                   00077100
        N3=70                                                                   00077200
        GO TO 16                                                                00077300
99      CONTINUE                                                                00077400
        IF(N1 .NE. NREAD) GO TO 152                                             00077500
        NFN1(1)=76                                                              00077600
        NFN1(2)=77                                                              00077700
        NFN1(3)=N2+5                                                            00077800
        NFN1(4)=78                                                              00077900
        NFN1(5)=0                                                               00078000
```

```
      JCNT=0                                                            00078100
100   CONTINUE                                                          00078200
      JSW=2                                                             00078300
      JCNT=JCNT+1                                                       00078400
      IF(NFN1(JCNT) .EQ. 0) GO TO 7                                     00078500
      JSW=5                                                             00078600
      N5=NFN1(JCNT)                                                     00078700
      GO TO 195                                                         00078800
101   CONTINUE                                                          00078900
C:SUBROUTINE STATEMENT. RESET DICTIONARY COUNT.                         00079000
      INTRY=0                                                           00079100
      IFRTRN=0                                                          00079200
      N2=N1                                                             00079300
      N1=2                                                              00079400
      N3=0                                                              00079500
      N5=8                                                              00079600
      GO TO 152                                                         00079700
102   CONTINUE                                                          00079800
      LBYTE(NREAD+1)=KARSET(46)                                         00079900
      N1=NREAD+3                                                        00080000
      N2=0                                                              00080100
      N4=12                                                             00080200
C: INSERT PROC.                                                         00080300
      GO TO 73                                                          00080400
103   CONTINUE                                                          00080500
C:GO COMMAND. SQUEEZE INTO GOTO.                                        00080600
      I=LBYTE(N1+1)                                                     00080700
      IF(MAP(I) .NE. 50) GO TO 152                                      00080800
      J=N1+2                                                            00080900
      DO 104 I=J,NREAD                                                  00081000
      LBYTE(I-1)=LBYTE(I)                                               00081100
104   CONTINUE                                                          00081200
C:CLOSE THE SPACE BETWEEN GO AND TO AND TRY AGAIN.                      00081300
      NREAD=NREAD-1                                                     00081400
      N1=LOOP-1                                                         00081500
      GO TO 45                                                          00081600
105   CONTINUE                                                          00081700
C:GOTO (NOW PROPERLY SQUEEZED).                                         00081800
      NGOTO=LOOP                                                        00081900
      N5=9                                                              00082000
      GO TO 46                                                          00082100
106   CONTINUE                                                          00082200
      IF(I .GT. 9) GO TO 107                                            00082300
C:SIMPLE GOTO. PREFIX THE LABEL WITH A C.                               00082400
      N2=0                                                              00082500
      N4=49                                                             00082600
      GO TO 73                                                          00082700
107   CONTINUE                                                          00082800
      IF(I .NE. 40) GO TO 152                                           00082900
C:COMPUTED GOTO. CHANGE TO SWITCH.                                      00083000
      N1=NGOTO                                                          00083100
      N2=LOOP                                                           00083200
      N3=2                                                              00083300
      N4=50                                                             00083400
      N5=9                                                              00083500
      GO TO 152                                                         00083600
108   CONTINUE                                                          00083700
C:MOVE THE ITEM FROM END INTO PARENS.                                   00083800
      NGOTO=NGOTO+12                                                    00083900
      JCNT=NREAD                                                        00084000
```

```
      JMRK=0                                                            00084100
109   CONTINUE                                                          00084200
      JCNT=JCNT-1                                                       00084300
      IF(JCNT .LE. LOOP) GO TO 192                                      00084400
      I=LBYTE(JCNT)                                                     00084500
      I=MAP(I)                                                          00084600
      IF(I .EQ. 41) GO TO 110                                           00084700
      IF(I .LE. 35) GO TO 105                                           00084800
      IF(JMRK .EQ. 0) JMRK=JCNT+1                                       00084900
      IF(I .NE. 45 .AND. I .NE. 50) GO TO 192                           00085000
      LBYTE(JCNT)=KARSET(50)                                            00085100
C:BLANK THE COMMA.                                                      00085200
      GO TO 109                                                         00085300
110   CONTINUE                                                          00085400
      IF(JMRK .EQ. 0 .OR. NREAD-JMRK .GT. 5)                            00085500
     1   GO TO 192                                                      00085600
      LOOP=JMRK                                                         00085700
      N2=26                                                             00085800
      N3=30                                                             00085900
      N4=4                                                              00086000
      GO TO 16                                                          00086100
111   CONTINUE                                                          00086200
      DO 112 I=JMRK,NREAD                                               00086300
      LBYTE(NGOTO)=LBYTE(I)                                             00086400
      LBYTE(I)=KARSET(50)                                               00086500
      NGOTO=NGOTO+1                                                     00086600
112   CONTINUE                                                          00086700
      NREAD=JCNT                                                        00086800
C:PREFIX ALL LABELS WITH A Q.                                           00086900
113   CONTINUE                                                          00087000
      J=0                                                               00087100
114   CONTINUE                                                          00087200
      NGOTO=NGOTO+1                                                     00087300
      IF(NGOTO .GE. NREAD) GO TO 164                                    00087400
      I=LBYTE(NGOTO)                                                    00087500
      IF(MAP(I) .GT. 9) GO TO 113                                       00087600
      IF(J .NE. 0) GO TO 114                                            00087700
C:NEXT LABEL TO GET Q PREFIX.                                           00087800
      I=NREAD                                                           00087900
115   CONTINUE                                                          00088000
      LBYTE(I+1)=LBYTE(I)                                               00088100
      I=I-1                                                             00088200
      IF(I .GE. NGOTO) GO TO 115                                        00088300
      LBYTE(NGOTO)=KARSET(26)                                           00088400
      NREAD=NREAD+1                                                     00088500
      J=1                                                               00088600
      GO TO 114                                                         00088700
116   CONTINUE                                                          00088800
C:IABS FUNCTION. BLANK FIRST LETTER.                                    00088900
      LBYTE(LOOP)=KARSET(50)                                            00089000
117   CONTINUE                                                          00089100
      IF(IEQ .EQ. 0) GO TO 46                                           00089200
C:CHECK FOR PRECEDING MINUS SIGN FOR '1*' FIX.                          00089300
      N3=LOOP                                                           00089400
      N4=37                                                             00089500
118   CONTINUE                                                          00089600
      N3=N3-1                                                           00089700
      I=LBYTE(N3)                                                       00089800
      IF(MAP(I) .EQ. 50) GO TO 118                                      00089900
      IF(MAP(I) .NE. N4) GO TO 46                                       00090000
```

```
      N4=N4+7                                                          00090100
      IF(N4 .EQ. 44) GO TO 118                                         00090200
C:PUT IN THE CORRECTION.                                               00090300
      JCNT=NREAD-N1                                                    00090400
      N1=LOOP                                                          00090500
      N4=N2+67                                                         00090600
      N2=0                                                             00090700
      N3=2                                                             00090800
      GO TO 151                                                        00090900
119   CONTINUE                                                         00091000
C:ALOG, SQRT, SIN, COS FUNCTION CALLS.                                 00091100
      IDPTH=IDPTH+1                                                    00091200
      IF(IDPTH .GT. 5) GO TO 192                                       00091300
      KINDST(IDPTH)=3                                                  00091400
      NFN1(IDPTH)=LOOP                                                 00091500
      NFN2(IDPTH)=N1+1                                                 00091600
      IF(N2 .EQ. 20) GO TO 75                                          00091700
C:CHANGE ALOG TO LN.                                                   00091800
      LBYTE(LOOP)=KARSET(50)                                           00091900
      LBYTE(N1-1)=KARSET(23)                                           00092000
      LBYTE(N1)=KARSET(50)                                             00092100
      NFN1(IDPTH)=LOOP+1                                               00092200
      NFN2(IDPTH)=N1                                                   00092300
      GO TO 75                                                         00092400
120   CONTINUE                                                         00092500
C:SIN, COS -- MAKE INTO PROCEDURE CALL.                                00092600
      JCNT=NREAD-N1                                                    00092700
      N1=N1+1                                                          00092800
      N2=0                                                             00092900
      N3=2                                                             00093000
      N4=55                                                            00093100
      GO TO 151                                                        00093200
121   CONTINUE                                                         00093300
C:END OF CALLING ARGUMENT FOR ALOG, SQRT, SIN, COS.                    00093400
      JCNT=NREAD-N1                                                    00093500
      N1=LSTPRN                                                        00093600
      N2=0                                                             00093700
      N3=2                                                             00093800
C:ADD ENDING CHARACTERS.                                               00093900
      N4=53                                                            00094000
      N5=12                                                            00094100
      GO TO 152                                                        00094200
122   CONTINUE                                                         00094300
      N1=NFN2(IDPTH)                                                   00094400
      N2=0                                                             00094500
      N3=2                                                             00094600
      N4=57                                                            00094700
C:INSERT (SHORT                                                        00094800
      N5=13                                                            00094900
      GO TO 152                                                        00095000
123   CONTINUE                                                         00095100
      N1=NFN1(IDPTH)                                                   00095200
      IDPTH=IDPTH-1                                                    00095300
      N2=0                                                             00095400
      N3=2                                                             00095500
      N4=52                                                            00095600
C:INSERT LONG(                                                         00095700
      GO TO 151                                                        00095800
124   CONTINUE                                                         00095900
      N1=NREAD-JCNT                                                    00096000
```

```
      GO TO 45                                                          00096100
125   CONTINUE                                                          00096200
C:SCANNING RIGHT OF THE FIRST CHARACTER.                                00096300
      IF(I .GT. 9) GO TO 127                                            00096400
C:CHECK THE NUMBER.                                                     00096500
126   CONTINUE                                                          00096600
      N4=4                                                              00096700
      GO TO 28                                                          00096800
127   CONTINUE                                                          00096900
      IF(I .GT. 35) GO TO 128                                           00097000
C:CHECK THE NAME.                                                       00097100
      N2=15                                                             00097200
      GO TO 53                                                          00097300
128   CONTINUE                                                          00097400
      IF(I .EQ. 38) GO TO 130                                           00097500
      IF(I .EQ. 39) GO TO 131                                           00097600
      IF(I .EQ. 40) GO TO 76                                            00097700
      IF(I .EQ. 41) GO TO 79                                            00097800
      IF(I .EQ. 44) IEQ=LOOP                                            00097900
      IF(I .NE. 42) GO TO 46                                            00098000
      IF(LOOP .GT. NREAD-3) GO TO 126                                   00098100
C:TEST FOR RELATIONAL.                                                  00098200
      N2=31                                                             00098300
      N3=38                                                             00098400
      GO TO 16                                                          00098500
129   CONTINUE                                                          00098600
C:RELATIONAL FOUND.  DROP PERIODS.                                      00098700
      N1=N1+1                                                           00098800
      I=LBYTE(N1)                                                       00098900
      IF(MAP(I) .NE. 42) GO TO 192                                      00099000
      LBYTE(N1)=KARSET(50)                                              00099100
      LBYTE(LOOP)=KARSET(50)                                            00099200
      IF(N2 .LE. 36) GO TO 45                                           00099300
      LBYTE(N1)=KARSET(40)                                              00099400
      LBYTE(LOOP)=KARSET(41)                                            00099500
      GO TO 45                                                          00099600
130   CONTINUE                                                          00099700
C:ASTERISK.  CHECK FOR DOUBLE.                                          00099800
      I=LBYTE(N1-1)                                                     00099900
      IF(MAP(I) .NE. 38) GO TO 46                                       00100000
      N5=83                                                             00100100
      GO TO 195                                                         00100200
131   CONTINUE                                                          00100300
      IF(IEQ .EQ. 0 .AND. KINDL .EQ. 0 .OR. IEQ .NE. 0                  00100400
     1    .AND. KINDR .EQ. 0) GO TO 46                                  00100500
C:ADD TRUNC FUNCTION TO FIX DIVIDE SCALING PROBLEM.                     00100600
      JCNT=NREAD-N1                                                     00100700
      JMRK=N1                                                           00100800
      N3=0                                                              00100900
      N4=0                                                              00101000
      N5=1                                                              00101100
132   CONTINUE                                                          00101200
      JMRK=JMRK-1                                                       00101300
      IF(JMRK .LE. IBGN) GO TO 192                                      00101400
      I=LBYTE(JMRK)                                                     00101500
      I=MAP(I)                                                          00101600
      IF(I*N5 .EQ. 50) GO TO 132                                        00101700
C:SKIP TRAILING BLANKS.                                                 00101800
      N5=0                                                              00101900
```

```
      IF(I .NE. 40) GO TO 134                                           00102100
C:OPEN PAREN.                                                           00102200
      N3=N3-1                                                           00102300
      IF(N3) 137,133,132                                                00102400
133   CONTINUE                                                          00102500
C:CHECK FURTHER FOR POSSIBLE FUNCTION.                                  00102600
      N4=JMRK                                                           00102700
      N5=1                                                              00102800
      GO TO 132                                                         00102900
134   CONTINUE                                                          00103000
      IF(I .NE. 41) GO TO 135                                           00103100
C:CLOSE PAREN.                                                          00103200
      IF(N5 .LT. 0) GO TO 137                                           00103300
      N3=N3+1                                                           00103400
      GO TO 132                                                         00103500
135   CONTINUE                                                          00103600
      IF(N3 .NE. 0 .OR. I .EQ. 42) GO TO 132                            00103700
C:CONTINUE FOR PERIOD OR WITHIN PARENS.                                 00103800
136   CONTINUE                                                          00103900
      IF(I .GT. 35) GO TO 137                                           00104000
C:INTO A NAME OR NUMBER.                                                00104100
      IF(N4 .NE. 0) N4=JMRK                                             00104200
      N5=-1                                                             00104300
      GO TO 132                                                         00104400
137   CONTINUE                                                          00104500
C:MARK BEGINNING OF TRUNC FIELD BY JMRK.                                00104600
      JMRK=JMRK+1                                                       00104700
      IF(N4 .NE. 0) JMRK=N4                                             00104800
      N2=0                                                              00104900
      N3=2                                                              00105000
      N4=81                                                             00105100
      N5=11                                                             00105200
      GO TO 152                                                         00105300
138   CONTINUE                                                          00105400
      N1=JMRK                                                           00105500
      N2=0                                                              00105600
      N3=2                                                              00105700
      N4=82                                                             00105800
      GO TO 151                                                         00105900
139   CONTINUE                                                          00106000
C:UNDECLARED ITEM NAME ERROR.                                           00106100
      N5=63                                                             00106200
      NUMERR=NUMERR+1                                                   00106300
140   CONTINUE                                                          00106400
      JSW=4                                                             00106500
      GO TO 156                                                         00106600
141   CONTINUE                                                          00106700
      JSW=2                                                             00106800
      GO TO 45                                                          00106900
142   CONTINUE                                                          00107000
C:ITEM DECLARATION.                                                     00107100
      JMRK=LOOP                                                         00107200
      JCNT=N1                                                           00107300
      IF(N1 .GE. NREAD) GO TO 146                                       00107400
      N5=11                                                             00107500
      GO TO 46                                                          00107600
143   CONTINUE                                                          00107700
      IF(I .EQ. 40) GO TO 144                                           00107800
      N1=JCNT                                                           00107900
      GO TO 146                                                         00108000
```

```
144     CONTINUE                                                        00108100
C:DIMENSIONED ITEM.                                                     00108200
        IF(KOMMON .EQ. 0) GO TO 139                                     00108300
        N5=12                                                           00108400
        GO TO 46                                                        00108500
145     CONTINUE                                                        00108600
        IF(I .LE. 9) GO TO 46                                           00108700
        IF(I .NE. 41) GO TO 152                                         00108800
C:CLOSE PAREN OF DIMENSIONED ITEM.                                      00108900
        JCNT=N1                                                         00109000
146     CONTINUE                                                        00109100
        IF(NTYPE .LT. 3) GO TO 45                                       00109200
C:OUTPUT THE DCL STATEMENT.                                             00109300
        DO 147 I=1,72                                                   00109400
        ICMPRE(I)=KARSET(50)                                            00109500
147     CONTINUE                                                        00109600
        N3=59                                                           00109700
        IF(N2 .EQ. 0) N3=60                                             00109800
C:IF N2 IS ZERO, THE ITEM IS FLOATING. OTHERWISE, IT IS INTEGER.        00109900
        IBYT1=MSGDR1(N3)                                                00110000
        IBYT2=MSGDR2(N3)                                                00110100
        INDEX=IWCMP+7                                                   00110200
        CALL LDBYTE                                                     00110300
        N3=12                                                           00110400
148     CONTINUE                                                        00110500
        ICMPRE(N3)=LBYTE(JMRK)                                          00110600
        N3=N3+1                                                         00110700
        JMRK=JMRK+1                                                     00110800
        IF(JMRK .LE. JCNT) GO TO 148                                    00110900
        N1=JCNT                                                         00111000
        N5=0                                                            00111100
        GO TO 140                                                       00111200
149     CONTINUE                                                        00111300
        IF(N1 .NE. NREAD) GO TO 152                                     00111400
        IF(IFRTRN .EQ. 0) GO TO 164                                     00111500
C:END STATEMENT. CHANGE TO QREND: END.                                  00111600
        N2=N1                                                           00111700
        N1=LOOP                                                         00111800
        N4=61                                                           00111900
        GO TO 73                                                        00112000
150     CONTINUE                                                        00112100
C:RETURN STATEMENT. CHANGE TO GOTO QREND.                               00112200
        IFRTRN=1                                                        00112300
        JCNT=NREAD-N1                                                   00112400
        N2=N1                                                           00112500
        N1=LOOP                                                         00112600
        N3=2                                                            00112700
        N4=62                                                           00112800
151     CONTINUE                                                        00112900
        N5=14                                                           00113000
152     CONTINUE                                                        00113100
C:REPLACE CHARACTERS IN LBYTE ARRAY.                                    00113200
C:N1--FIRST REPLACEMENT POSITION.                                       00113300
C:N2--SECOND REPLACEMENT POSITION.                                      00113400
C:N3--0  DELETE, 1  NUMBER REPLACEMENT, 2  STRING REPLACEMENT.          00113500
C:N4--NUMBER (OR MESSAGE NUMBER), 0 LABEL IF MINUS.                     00113600
C:N5--RETURN CODE.                                                      00113700
        IF(N2 .LT. N1) N2=N1-1                                          00113800
        IF(NREAD .LT. N2) NREAD=N2                                      00113900
        IF(N3 .NE. 2) GO TO 153                                         00114000
```

```
C:REPLACEMENT STRING.                                                  00114100
      IBYT1=MSGDR1(N4)                                                 00114200
      IBYT2=MSGDR2(N4)                                                 00114300
      INDEX=INCMP                                                      00114400
      CALL LDBYTE                                                      00114500
      N3=1                                                             00114600
      N4=IBYT2-IBYT1+1                                                 00114700
      GO TO 157                                                        00114800
153   CONTINUE                                                         00114900
      IF(N3 .EQ. 0) GO TO 155                                          00115000
C:REPLACEMENT NUMBER.                                                  00115100
      N3=MAXLIN                                                        00115200
      J=N4                                                             00115300
      N4=IABS(N4)                                                      00115400
154   CONTINUE                                                         00115500
      IBYT1=N4/10                                                      00115600
      IBYT2=N4-10*IBYT1                                                00115700
      N4=IBYT1                                                         00115800
      ICMPRE(N3)=KARSET(IBYT2)                                         00115900
      IF(N4 .EQ. 0) GO TO 156                                          00116000
      N3=N3-1                                                          00116100
      GO TO 154                                                        00116200
155   CONTINUE                                                         00116300
      N4=-1                                                            00116400
      GO TO 157                                                        00116500
156   CONTINUE                                                         00116600
      N4=MAXLIN                                                        00116700
      IF(J .GE. 0) GO TO 157                                           00116800
      N3=N3-1                                                          00116900
      ICMPRE(N3)=KARSET(26)                                            00117000
C:PREFIX THE Q BEFORE THE LABEL.                                       00117100
157   CONTINUE                                                         00117200
      I=N1-N2-N3+N4                                                    00117300
      IBYT1=NREAD                                                      00117400
      IF(I) 158,161,159                                                00117500
158   CONTINUE                                                         00117600
C:REDUCE SPACE.                                                        00117700
      N2=N2+1                                                          00117800
      IF(N2 .GT. NREAD) GO TO 160                                      00117900
      J=N2+I                                                           00118000
      LBYTE(J)=LBYTE(N2)                                               00118100
      GO TO 158                                                        00118200
159   CONTINUE                                                         00118300
C:INCREASE SPACE.                                                      00118400
      J=IBYT1+I                                                        00118500
      LBYTE(J)=LBYTE(IBYT1)                                            00118600
      IBYT1=IBYT1-1                                                    00118700
      IF(IBYT1 .GT. N2) GO TO 159                                      00118800
160   CONTINUE                                                         00118900
C:SPACE NOW CORRECT.                                                   00119000
      NREAD=NREAD+I                                                    00119100
161   CONTINUE                                                         00119200
      IF(N3 .EQ. 0) GO TO 163                                          00119300
162   CONTINUE                                                         00119400
      LBYTE(N1)=ICMPRE(N3)                                             00119500
      N1=N1+1                                                          00119600
      N3=N3+1                                                          00119700
      IF(N3 .LE. N4) GO TO 162                                         00119800
163   CONTINUE                                                         00119900
      IF(I .GE. 0) GO TO (49,43,81,87                                  00120000
```

```
     1    ,69,94,164,102,108,47                                                  00120100
     2    ,138,122,123,124),NS                                                   00120200
          J=NREAD-1                                                              00120300
          LBYTE(J)=KARSET(50)                                                    00120400
          I=I+1                                                                  00120500
          GO TO 162                                                              00120600
164       CONTINUE                                                              00120700
C:APPEND SEMI-COLON AND SEQUENCE NUMBER AND OUTPUT THE LINE.                     00120800
          NREAD=NREAD+1                                                          00120900
          LBYTE(NREAD)=KARSET(47)                                                00121000
165       CONTINUE                                                              00121100
          IF(NTYPE .GE. 3) GO TO 185                                             00121200
166       CONTINUE                                                              00121300
          I=LBYTE(1)                                                             00121400
          IF(MAP(I) .EQ. 5C) GO TO 168                                           00121500
          NREAD=NREAD+1                                                          00121600
          J=NREAD                                                                00121700
167       CONTINUE                                                              00121800
          J=J-1                                                                  00121900
          LBYTE(J+1)=LBYTE(J)                                                    00122000
          IF(J .GT. 1) GO TO 167                                                 00122100
          LBYTE(1)=KARSET(50)                                                    00122200
          IF(MAP(I) .GT. 5) GO TO 168                                            00122300
C:ADD THE Q PREFIX.                                                              00122400
          LBYTE(1)=KARSET(26)                                                    00122500
          GO TO 166                                                              00122600
168       CONTINUE                                                              00122700
          IF(MAP(I) .EQ. 35 .OR. NREAD .LE. 6) GO TO 193                         00122800
          I=3                                                                    00122900
          J=3                                                                    00123000
169       CONTINUE                                                              00123100
          I=I+1                                                                  00123200
170       CONTINUE                                                              00123300
          J=J+1                                                                  00123400
          IF(J-NREAD) 171,172,173                                                00123500
171       CONTINUE                                                              00123600
          IF(LBYTE(J) .EQ. KARSET(50) .AND.                                      00123700
     1    LBYTE(J+1) .EQ. KARSET(50)) GO TO 170                                  00123800
172       CONTINUE                                                              00123900
          IF(I .EQ. J) GO TO 165                                                 00124000
          LBYTE(I)=LBYTE(J)                                                      00124100
          LBYTE(J)=KARSET(50)                                                    00124200
          GO TO 169                                                              00124300
173       CONTINUE                                                              00124400
          NREAD=I-1                                                              00124500
174       CONTINUE                                                              00124600
          IF(NREAD .LE. 70) GO TO 193                                            00124700
          K=70                                                                   00124800
175       CONTINUE                                                              00124900
          J=LBYTE(K-1)                                                           00125000
          J=MAP(J)                                                               00125100
          IF(J .GE. 36 .AND. J .LE. 41 .OR.                                      00125200
     1    J .EQ. 45 .OR. J .EQ. 50) GO TO 176                                    00125300
          K=K-1                                                                  00125400
          IF(K .EQ. 0) GO TO 192                                                 00125500
          GO TO 175                                                              00125600
176       CONTINUE                                                              00125700
          DO 177 I=1,72                                                          00125800
          ICMPRE(I)=KARSET(50)                                                   00125900
177       CONTINUE                                                              00126000
```

```
      J=7                                                               00126100
      DO 178 I=K,NREAD                                                  00126200
      ICMPRE(J)=LBYTE(I)                                                00126300
      LBYTE(I)=KARSET(50)                                               00126400
      J=J+1                                                             00126500
178   CONTINUE                                                          00126600
      NREAD=NREAD-K+7                                                   00126700
179   CONTINUE                                                          00126800
C:CHECK AND DON'T OUTPUT A BLANK LINE.                                  00126900
      DO 180 I=1,72                                                     00127000
      IF(LBYTE(I) .NE. KARSET(50) .AND.                                 00127100
     1    LBYTE(I) .NE. LSTCHK) GO TO 181                               00127200
180   CONTINUE                                                          00127300
      GO TO 183                                                         00127400
181   CONTINUE                                                          00127500
C:ATTACH LINE NUMBER.                                                   00127600
      LINE=LINE+1                                                       00127700
      LBYTE(79)=KARDUM                                                  00127800
      LBYTE(80)=KARDUM                                                  00127900
      I=78                                                              00128000
      N=LINE                                                            00128100
182   CONTINUE                                                          00128200
      J=N/10                                                            00128300
      K=N-10*J                                                          00128400
      N=J                                                               00128500
      LBYTE(I)=KARSET(K)                                                00128600
      I=I-1                                                             00128700
      IF(I .GE. 73) GO TO 182                                           00128800
      IBYT1=IBUFF                                                       00128900
      IBYT2=IBUFF+75                                                    00129000
      INDEX=IWLBYT                                                      00129100
      CALL SBYTE                                                        00129200
      GO TO 3                                                           00129300
183   CONTINUE                                                          00129400
      IF(IERR .NE. 0) GO TO 187                                         00129500
      IF(NREAD .EQ. 0) GO TO 185                                        00129600
      J=NREAD                                                           00129700
      IF(J .LT. 72) J=72                                                00129800
      DO 184 I=1,J                                                      00129900
      LBYTE(I)=ICMPRE(I)                                                00130000
184   CONTINUE                                                          00130100
      LSTCHK=KARSET(50)                                                 00130200
      GO TO 174                                                         00130300
185   CONTINUE                                                          00130400
      IERR=0                                                            00130500
      IF(JSW .EQ. 2) GO TO 7                                            00130600
      NREAD=IREAD                                                       00130700
      DO 186 I=1,MAXLIN                                                 00130800
      LBYTE(I)=ICMPRE(I)                                                00130900
186   CONTINUE                                                          00131000
      GO TO(192,192,50,141,100,1),JSW                                   00131100
187   CONTINUE                                                          00131200
C:ERROR LINE. PUT OUT LINE OF //////                                    00131300
      NUMERR=NUMERR+1                                                   00131400
      DO 188 I=1,72                                                     00131500
      LBYTE(I)=KARSET(35)                                               00131600
188   CONTINUE                                                          00131700
      IERR=0                                                            00131800
      GO TO 193                                                         00131900
189   CONTINUE                                                          00132000
```

```
      IF(NUMERR .EQ. 0) GO TO 154                                       00132100
C:END. PUT OUT SUMMARY.                                                 00132200
      DO 190 I=1,80                                                     00132300
      LBYTE(I)=KARSET(50)                                               00132400
190   CONTINUE                                                          00132500
      IBYT1=MSGDR1(65)                                                  00132600
      IBYT2=MSGDR2(65)                                                  00132700
      INDEX=IWLBYT                                                      00132800
      CALL LDBYTE                                                       00132900
      K=50                                                              00133000
191   CONTINUE                                                          00133100
      I=NUMERR/10                                                       00133200
      J=NUMERR-10*I                                                     00133300
      NUMERR=I                                                          00133400
      LBYTE(K)=KARSET(J)                                                00133500
      K=K-1                                                             00133600
      IF(I .NE. 0) GO TO 191                                            00133700
      GO TO 193                                                         00133800
192   CONTINUE                                                          00133900
C:CANNOT CRACK. PUT OUT ORIGINAL LINE AND //////                        00134000
      JSW=2                                                             00134100
      IF(NTYPE .GE. 3) GO TO 7                                          00134200
      IBYT1=IBUFF                                                       00134300
      IBYT2=IBUFF+71                                                    00134400
      INDEX=IWLBYT                                                      00134500
      CALL LDBYTE                                                       00134600
      IERR=1                                                            00134700
193   CONTINUE                                                          00134800
      NREAD=0                                                           00134900
      GO TO 179                                                         00135000
194   CONTINUE                                                          00135100
      IF(NTYPE .GE. 3) GO TO 1                                          00135200
      JSW=6                                                             00135300
      N5=80                                                             00135400
195   CONTINUE                                                          00135500
      IF(NTYPE .GE. 3) GO TO 7                                          00135600
C:OUTPUT LINE IN ICMPRE IF N5 EQ ZERO OR IN MSGDR(N5)                   00135700
C:IF N5 NOT ZERO. DO NOT CHANGE LBYTE.                                  00135800
196   CONTINUE                                                          00135900
      DO 197 I=1,MAXLIN                                                 00136000
      J=LBYTE(I)                                                        00136100
      LBYTE(I)=ICMPRE(I)                                                00136200
      ICMPRE(I)=J                                                       00136300
      IF(N5 .GT. 0) LBYTE(I)=KARSET(50)                                 00136400
197   CONTINUE                                                          00136500
      IREAD=NREAD                                                       00136600
      IF(N5 .EQ. 0) GO TO 153                                           00136700
      IBYT1=MSGDR1(N5)                                                  00136800
      IBYT2=MSGDR2(N5)                                                  00136900
      INDEX=IWLBYT+2                                                    00137000
      IF(N5 .EQ. 80) INDEX=IWLBYT                                       00137100
      CALL LDBYTE                                                       00137200
      GO TO 193                                                         00137300
198   CONTINUE                                                          00137400
      STOP                                                              00137500
      END                                                               00137600
```